GACIAC
GUIDANCE & CONTROL
INFORMATION ANALYSIS CENTER

GACIAC SOAR-95-01

MODERN CONTROL THEORY
State-of-the-Art Review

Dr. Robert J. Heaston, Editor

Published by:
IIT Research Institute/GACIAC
Guidance and Control Information Analysis Center
10 West 35th Street
Chicago, Illinois 60616-3799

December 1995

— Sponsored by the Defense Technical Information Center since 1977 —

GACIAC SOAR-95-01 **DECEMBER 1995**

NOTICES

State-of-the-Art Review. This state-of-the-art review has been published by the Guidance and Control Information Analysis Center (GACIAC) as a service to the defense community. GACIAC is a DoD Information Analysis Center, administered by the Defense Technical Information Center, operated by IIT Research Institute under Contract No. DLA-900-93-D-0011. GACIAC is funded by DTIC, ARPA and U.S. Army, U.S. Navy, and U.S. Air Force Laboratories/Controlling Activities having an interest in weapon guidance and control. The Contracting Officer is Ms. Cheryl Montoney, DESC, Dayton, Ohio. The Contracting Officer's Technical Representative is Mr. Chalmer D. George, and the Alternate Representative is Mr. Howard C. Race, U.S. Army Missile Command, AMC Smart Weapons Management Office, Attn: AMSMI-SW, Redstone Arsenal, Alabama 35898-5222. The Director of GACIAC is Dr. Robert J. Heaston.

Special tasks to conduct analyses, develop simulations, design special equipment, and provide engineering support in the area of weapon guidance and control for DoD agencies and laboratories are authorized under the GACIAC contract. Such tasks are intended to provide rapid and specialized support to the DoD community in areas where GACIAC has the required background and experience to accomplish the required results in a short time period.

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188

1. AGENCY USE ONLY (*Leave blank*)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	December 1995	State-of-the-Art Review; Dec 1990-Sep 1995

4. TITLE AND SUBTITLE Modern Control Theory	5. FUNDING NUMBERS DoD-DLA900-93-D-0011 PE 65802
6. AUTHOR(S) Dr. R.J. Heaston, Editor	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) IIT Research Institute/GACIAC 10 West 35th Street Chicago, IL 60616-3799	8. PERFORMING ORGANIZATION REPORT NUMBER SOAR 95-01
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) US Army Missile Command AMC Smart Weapons Management Office Attn: AMSMI-SW Redstone Arsenal, AL 35898-5222	10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

This document is available only from GACIAC, IIT Research Institute, 10 West 35th Street, Chicago, IL 60616-3799 (410948)

12a. DISTRIBUTION/AVAILABILITY STATEMENT	12b. DISTRIBUTION CODE
Approved for public release; distribution unlimited.	A

13. ABSTRACT (*Maximum 200 words*)

Modern control theory is based on abstract mathematical concepts and its development uses a system of notation and terminology largely incomprehensible to engineers and managers not skilled in the art. As a body of knowledge, modern control theory encompasses all of classical control system design, augmented with computational techniques largely developed over the past two decades. Although these techniques are highly mathematical in nature, a knowledge of their general approaches and some familiarity with their results is necessary to appreciate and comprehend their intended applications. This review attempts to explain the concepts, advantages, and limitations of modern control theory in layman's language to the extent possible.

This GACIAC State-of-the-Art Review (SOAR) focuses on the application of selected concepts and mathematical tools drawn from modern control theory to the design and development of weapon guidance and control systems. This review addresses the basic concepts of this technology, their present application, and their future potential.

14. SUBJECT TERMS	15. NUMBER OF PAGES
Control, Control theory, Dynamic systems, Stochastic control, Adaptive control, Optimal control, Optimization, Game theory, Differential games, Filters, Kalman filters, Estimation, and Guided Missiles.	384
	16. PRICE CODE $150

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED/UNLIMITED	UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18
298-102

GACIAC SOAR-95-01 NOVEMBER 1995

MODERN CONTROL THEORY

Dr. Robert J. Heaston, Editor

Prepared
and
Published by:

IIT Research Institute/GACIAC
Guidance and Control Information Analysis Center
10 West 35th Street
Chicago, Illinois 60616-3799

GACIAC - A DoD Information Analysis Center
Sponsored by the Defense Technical Information Center
Operated by IIT Research Institute, 10 West 35th Street, Chicago, Illinois 60616-3799
DoD Technical Sponsor-Joint Service Guidance and Control Committee
Members from OSD, Army, Navy, Air Force and ARPA

PREFACE

Demands for improved performance of tactical guided weapons expected to operate in complex battle environments lead to the investigation and application of guidance and control systems based on modern control theory.

Modern control theory deals with the analysis and synthesis of systems and devices for the control of complex multi-input multi-output systems. Modern control theory, despite its apparent mathematical complexity, provides a unified approach to solving a wide variety of guidance and control analysis, design, and optimization problems. The application of modern control theory to the development of tactical guided weapons is a high interest, high potential technology.

Modern control theory is based on abstract mathematical concepts and its development uses a system of notation and terminology largely incomprehensible to engineers and managers not skilled in the art. As a body of knowledge, modern control theory encompasses all of classical control system design, augmented with computational techniques largely developed over the past two decades. Although these techniques are highly mathematical in nature, a knowledge of their general approaches and some familiarity with their results is necessary to appreciate and comprehend their intended applications. This review attempts to explain the concepts, advantages, and limitations of modern control theory in layman's language to the extent possible.

This GACIAC State-of-the-Art Review (SOAR) focuses on the application of selected concepts and mathematical tools drawn from modern control theory to the design and development of tactical weapon guidance and control systems. These tools include state-variable modeling and analysis, system identification, state and parameter estimation, optimization and optimal control, stochastic control, differential games, and adaptive control. This review addresses the basic concepts of these technologies, their present application, and their future potential. Readers are encouraged to pursue the topics presented in more detail, and to seek new applications for modern control theory in future tactical weapon systems.

Except for a few reports and conferences, GACIAC has mostly concentrated on the guidance partner of guidance and control. This review addresses the mostly silent partner. Dr. Donald S. Szarkowicz wrote most of the rough draft that was used as a basis of this report; he left IIT Research Institute in 1992 to change careers. His rough notes and drafts on discs were edited, restructured and amended to produce this report. Mrs. Susan Garrison, Ms. Karen Kozola, and Mrs. Toni Cavalieri processed and assembled the text and figures. I performed the editing and am responsible for any errors and omissions.

Robert J. Heaston
GACIAC Director

Modern Control Theory - State of the Art Review

TABLE OF CONTENTS

Chapter		Page
	Report Documentation Page	*i*
	Title Page	*ii*
	Preface	*iii*
	Table of Contents	*iv*
	List of Figures and Tables	*ix*
1.	**Introduction**	1-1
	1.1 Assessment of Need	1-1
	1.2 Report Structure and Content	1-2
2.	**Classical Control Theory**	2-1
	2.1 Introduction	2-1
	2.2 System Representations	2-1
	2.3 Analysis of Continuous-Time Control Systems	2-3
	2.4 Analysis of Discrete-Time Control Systems	2-4
	2.5 Reasons for Using Feedback as a Means of Obtaining Control	2-6
	2.6 Classical Closed-Loop Control System Performance Measures	2-7
	2.6.1 Stability	2-7
	2.6.2 Steady-State Accuracy	2-11
	2.6.3 Satisfactory Transient Response	2-12
	2.6.4 Satisfactory Frequency Response	2-14
	2.7 Classical Methods for Improving Performance	2-14
	2.8 Classical Performance Measures and Analytical Methods	2-16
	2.9 Continuous-Time Control System Design Example	2-16
	2.10 Discrete-Time Control System Design Example	2-25
	2.11 Summary	2-29
	References	2-29
3.	**Modern Control Theory**	3-1
	3.1 System Modeling	3-1
	3.2 The Development of State Variable Methods	3-6
	3.3 Stability Analysis of Dynamic Systems	3-7
	3.4 Optimal Control	3-8
	3.5 The Linear Quadratic Regulator	3-10
	3.6 A Linear Quadratic Regulator Example	3-12
	3.7 State Variable Feedback and Controllability	3-14
	3.8 Pole Placement or Eigenvalue Assignment	3-17
	3.9 The Use of Output Feedback	3-20
	3.10 State Variable Observers	3-22
	3.11 Summary	3-27
	References	3-28

4. Dynamic Systems 4-1

4.1 System Concepts 4-1
4.2 Dynamic System Problems 4-3
4.3 Modeling Dynamic Systems 4-4
4.4 Summary 4-6
References 4-7

5. System Identification 5-1

5.1 The Basic Identification Problem 5-1
5.2 Identification Methods in Modern Control Theory 5-6
5.3 Recursive Methods of System Identification 5-10
5.4 Least-Squares Methods 5-11
5.5 Least-Squares System Identification 5-12
5.6 Least-Squares System Identification Numerical Example 5-15
5.7 Maximum Likelihood Method 5-17
5.8 Maximum Likelihood Estimation of System Parameters 5-22
5.9 A Steepest Descent Algorithm for Maximum Likelihood Parameter Estimation 5-24
5.10 Summary 5-26
References 5-26

6. The Kalman Filter 6-1

6.1 The Basic Estimation Problem 6-1
6.1.1 Bayesian Parameter Estimation 6-3
6.1.2 Non-Bayesian Parameter Estimation 6-4
6.2 Nonlinear Estimation 6-5
6.3 Kalman Filtering and Applications 6-5
6.3.1 Kalman Filter Algorithm 6-6
6.3.2 Split Kalman Filter 6-10
6.3.3 Extended Kalman Filter 6-11
6.3.4 Noise and Process Modeling 6-18
6.3.5 Kalman Filter Divergence 6-18
6.3.6 Kalman Filter Design Considerations 6-19
6.3.7 Application to Target Tracking 6-20
6.4 Summary 6-22
References 6-23

7. Adaptive Control 7-1

7.1 Introduction 7-1
7.2 Applications of Adaptive Control 7-3
7.3 Overview of Adaptive Control Methods 7-4
7.4 Model Reference Adaptive Control Systems 7-8
7.4.1 Model Reference Adaptive Control Design Parameters 7-8
7.4.2 Model Reference Adaptive Control System Design Example 7-9
7.5 Self-Tuning Regulator Adaptive Control Systems 7-13
7.5.1 Self-Tuning Regulator Design Principles 7-14
7.5.2 Self-Tuning Regulator Design Example 7-16

7.6 Comparing Model Reference and Self-Tuning Adaptive Control Systems 7-18
7.7 Summary 7-20
References 7-21

8. Mathematical Optimization 8-1

8.1 Static Optimization Problem 8-1
8.2 Linear Programming 8-3
8.3 Nonlinear Programming 8-7
8.3.1 Gradient Descent Method 8-7
8.3.2 Conjugate Gradient Method 8-9
8.3.3 Newton's Method 8-10
8.3.4 Quasi-Newton Methods 8-10
8.4 Calculus of Variations 8-11
8.5 Dynamic Optimization 8-15
8.5.1 Dynamic Systems 8-16
8.5.2 Controlled Systems 8-16
8.5.3 Feedback and Open-Loop Control 8-17
8.6 Dynamic Programming 8-18
8.7 Summary 8-23
References 8-25

9. Optimal Control Theory 9-1

9.1 Introduction 9-1
9.2 Development of Optimal Control Theory 9-2
9.3 Pontryagin's Maximum Principle 9-4
9.4 Open-Loop and Closed-Loop Optimal Control 9-13
9.5 Performance Measures and Optimal Control Problems 9-14
9.5.1 Minimum Time Problem 9-15
9.5.2 Terminal Control Problem 9-15
9.5.3 Minimum Control Effort Problems 9-16
9.5.4 Tracking and Regulator Problems 9-16
9.5.5 Linear Quadratic Optimal Control Problem 9-18
9.6 Summary 9-22
References 9-23

10. Singular Perturbation Methods 10-1

10.1 Introduction to the Singular Perturbation Method 10-1
10.2 Stability Analysis of Singularly Perturbed Systems 10-6
10.3 Optimal Control for Singularly Perturbed Systems 10-7
10.4 Application to Linear Quadratic Optimal Control 10-9
10.5 Summary 10-10
References 10-11

11. Stochastic Control 11-1

11.1 Formulating Stochastic Control Problems 11-1
11.2 State Transition Tables and Diagrams 11-5
11.3 Feedback in Stochastic Control 11-7

11.4 Discrete-Time Stochastic Optimal Control 11-9
11.5 Linear Quadratic Gaussian Control Problem 11-13
11.6 The Separation Principle 11-14
11.7 Design of Suboptimal Stochastic Controllers 11-15
11.8 Extremal Control Systems 11-22
11.9 Summary 11-22
References 11-23

12. Differential Game Theory 12-1

12.1 Introduction 12-1
12.2 Discrete Game Theory 12-2
12.3 Continuous Games 12-3
12.4 Prototype Differential Games 12-4
12.5 The Four Elements of a Differential Game 12-5
12.5.1 State Transition Mechanism 12-5
12.5.2 Termination Condition 12-6
12.5.3 Player Performance Measures 12-6
12.5.4 Admissible Control Strategies 12-7
12.6 Differential Games With Two Players 12-7
12.7 Numerical Solution of Two-Player Differential Games 12-8
12.8 Linear-Quadratic Pursuit-Evasion Differential Games 12-11
12.9 Guidance Law for Three-Dimensional Target Interception 12-17
12.10 Proportional Navigation 12-18
12.11 Summary 12-21
References 12-21

13. Robustness and Sensitivity 13-1

13.1 Definitions 13-1
13.2 The Design of Robust Control Systems 13-2
13.3 An Example of Robust Controller Design 13-5
13.4 Sensitivity Analysis 13-10
13.5 Summary 13-13
References 13-13

14. Precision Guided Munitions 14-1

14.1 Overview Chapters 14-1
14.2 Classification of PGMs by Launcher-Target Locations 14-3
14.3 Classifications of PGMs by Target Type 14-3
14.4 Classification of PGMs by Sensor Operation 14-4
14.5 Classification of PGMs by Munition Type 14-8
14.6 Classification of PGMs by Formal Military Designations 14-8
14.7 Classification of PGMs by Capabilities 14-9
14.8 Classification of PGMs by Range 14-11
14.9 Classification of PGMs by Direct Guidance 14-11
14.10 Classification of PGMs by Guidance Laws 14-16
14.11 Future of PGMs 14-19
14.12 Summary 14-22
References 14-22

15. Applications of Control Theory to PGMS 15-1

15.1 PGM Simulation 15-1
15.2 Theory of Disturbance Accommodating Controllers 15-4
15.2.1 Waveform Mode Description of Realistic Disturbances 15-8
15.2.2 Waveform-Mode Versus Statistical Representation of Disturbances 15-10
15.2.3 State Variable Modeling of Waveform Structure 15-11
15.2.4 A Waveform Description of Unfamiliar Disturbances 15-15
15.2.5 The Design of Disturbance Accommodating Controllers for Stabilization, Regulation and Tracking Control Problems 15-16
15.2.6 State Observers for Signals with Waveform Structure 15-19
15.3 Closed-Loop System Analysis Using Lyapunov Stability 15-20
15.3.1 Homing Missile Guidance with Angle-Only Measurements 15-21
15.3.2 Conclusions About Lyapunov Functions 15-24
15.4 Numerical Methods for the Guidance and Control of Air-to-Air Missile 15-25
15.4.1 Requirements for an Air-to-Air Missile 15-26
15.4.2 An Overview of Two-Person, Zero-Sum, Differential Games 15-29
15.4.3 Kinematic Equations of Motion 15-32
15.4.4 Solution of the Deterministic Nonlinear Open-Loop Saddle Point Problem 15-33
15.4.5 Solution to the Nonlinear State Estimation Problem 15-35
15.5 Optimal Guidance of Homing Missiles 15-36
15.5.1 The State Variable Modeling Problem 15-38
15.5.2 The Homing Missile Guidance Problem 15-43
15.5.3 Implementation of Proportional Bang-Bang Guidance 15-45
15.6 A Study of Maximum Information Trajectories for Homing Missile Guidance 15-46
15.7 Summary 15-49
References 15-50

16. Gun Fire Control 16-1

16.1 Definitions 16-1
16.2 Gun Fire Control Processes 16-4
16.3 Gun Fire Control Configurations 16-5
16.4 Modern Control Theory Concepts 16-9
16.5 Summary 16-13
Reference 16-15

17. An Assessment of Modern Control Theory 17-1

17.1 An Assessment of Air-to-Air Missile Guidance and Control Technology 17-1
17.2 Target State Estimation 17-2
17.3 State Variable Filtering and Estimation Techniques 17-3
17.4 Advanced Guidance Laws 17-4
17.5 Bank-to-Turn Autopilot Design 17-5
17.6 Summary 17-7
References 17-9

LIST OF FIGURES AND TABLES

Figure/Table Number — Page

Chapter 1 - Introduction (No Figures)

Chapter 2 - Classical Control Theory

Figure 2-1 Single-input, single-output continuous-time feedback control system 2-2
Figure 2-2 Single-input, single-output discrete-time control system 2-2
Figure 2-3 Open-loop system components 2-17
Figure 2-4 Closed-loop servomechanism 2-18
Figure 2-5a Frequency response in decibels for a Bode plot of an uncompensated open-loop system 2-19
Figure 2-5b Phase angle of the frequency response in degrees for a Bode plot of an uncompensated open-loop system 2-19
Figure 2-6 Bode plot for an uncompensated open-loop system with a gain of 1200 2-21
Figure 2-7 Bode plot for a compensated, gain adjusted (gain=1200) open-loop system 2-23
Figure 2-8 Transient response of a compensated closed-loop system for unit step and ramp inputs 2-24
Figure 2-9 Compensated closed-loop system with a gain of 1200 2-25
Figure 2-10 Discrete-time closed-loop control system 2-26
Figure 2-11 Discrete-time closed-loop control system with a gain of 1200 2-27
Figure 2-12 Transient response of a discrete-time closed-loop control system for a sample time of 5 milliseconds 2-28
Table 2-1 Summary of Steady-State Errors 2-12

Chapter 3 - Modern Control Theory

Figure 3-1 Particle motion in the vertical plane 3-3
Figure 3-2 Block diagram of an unstable dynamic system 3-12
Figure 3-3 Stabilizing feedback control system 3-14
Figure 3-4 Uncontrollable dynamic system 3-16
Figure 3-5 Controllable dynamic system 3-17
Figure 3-6 Unstable open-loop system ($s_1=+3$, $s_2=-3$) 3-18
Figure 3-7 System stabilized by state variable feedback ($s_1=-1.0$, $s_2=0.5$) 3-20
Figure 3-8 Unstable closed-loop system with output feedback ($s_1=-1.0$, $s_2=+3.0$) 3-23
Figure 3-9 System stabilized by state variable feedback and use of full-order observer. 3-25
Figure 3-10 System stabilized by state variable feedback and use of reduced-order observer. 3-27

Chapter 4 - Dynamic Systems

Figure 4-1 Generalized system schematic 4-1

Chapter 5 - System Identification

Figure 5-1 System identification procedure 5-3
Figure 5-2 Transient response of a second-order system 5-4
Figure 5-3 Step response of a control system 5-5
Figure 5-4 Percent overshoot and peak time versus damping ratio for a second-order dynamic system 5-6
Figure 5-5 Single-input, single-output discrete linear time-invariant system 5-8
Figure 5-6 Single-input, single-output, time-invariant linear system 5-15

Chapter 6 - The Kalman Filter

Figure 6-1 Discrete-time, time variant stochastic system with a Kalman filter. 6-6
Figure 6-2 Kalman filter performance with an input disturbance w(k). 6-9
Figure 6-3 Kalman filter performance with a sine sequence control input. 6-10
Figure 6-4 Two-dimensional pursuit of a target by a surface-to-air missile. 6-14
Figure 6-5 Extended Kalman filter behavior for the example in Figure 6-4. 6-17

Chapter 7 - Adaptive Control

Figure 7-1 Stabilized double integrator. 7-2
Figure 7-2 Basic adaptive control system configuration. 7-3
Figure 7-3 System with gain scheduling. 7-5
Figure 7-4 Model reference adaptive control system. 7-6
Figure 7-5 Self-tuning regulator adaptive system. 7-6
Figure 7-6 Adaptive prediction and parameter estimation. 7-7
Figure 7-7 Transient, step responses of a time-invariant reference model and a time varying dynamic system. 7-12
Figure 7-8 Time-varying parameters b. 7-12
Figure 7-9 Time-varying gains. 7-13
Figure 7-10 Parameter estimates for a self-tuning regulator. 7-18
Figure 7-11 Transient response and control input for self-tuning regulator. 7-19

Chapter 8 - Mathematical Optimization

Figure 8-1 Multistage decision process 8-18
Figure 8-2 The brachistochrone problem 8-20
Figure 8-3 Dynamic programming solution of system states versus decision stages 8-22
Figure 8-4 Dynamic programming solution in x-y coordinates 8-24
Figure 8-5 Dynamic programming solution for minimum travel time for a brachistochrone problem 8-24

Chapter 9 - Optimal Control Theory

Figure 9-1 Parabolic Pontryagin solutions to a bang-bang control strategy. 9-8
Figure 9-2 Structure of the time-optimal controller. 9-9
Figure 9-3 Switching functions for time-optimal state trajectories. 9-10
Figure 9-4 Time-optimal switching function. 9-11
Figure 9-5 Optimal state and control variable trajectories. 9-21
Figure 9-6 Time-varying feedback gain. 9-21
Figure 9-7 State and control variable trajectories using suboptimal steady-state feedback gain. 9-22

Chapter 10 - Singular Perturbation Methods (No Figures)

Chapter 11 - Stochastic Control

Figure 11-1 A continuous-time stochastic process. 11-3
Figure 11-2 A discrete-time stochastic process. 11-4
Figure 11-3 Random process state trajectories. 11-5
Figure 11-4 Discrete-time stochastic process that models noise input and measurement errors as a sequence of random variables. 11-10
Figure 11-5 Optimal stochastic controller. 11-13

Figure 11-6 Linear-quadratic-Gaussian state variable trajectories 11-18
Figure 11-7 Linear-quadratic-Gaussian optimal control action 11-19
Figure 11-8 Time-varying gain resulting from linear-quadratic-Gaussian control 11-19
Figure 11-9 Effect of increased noise on linear-quadratic-Gaussian state variable trajectories . 11-20
Figure 11-10 Effect of increased noise on linear-quadratic-Gaussian control action. 11-21
Figure 11-11 Effect of process state trajectories on the performance of a stochastic optimal controller. 11-21

Table 11-1 State transition probabilities for a stochastic optimal control system 11-6

Chapter 12 - Differential Game Theory

Figure 12-1 Prototype differential game. 12-5
Figure 12-2 Linear-quadratic differential game trajectories. 12-16
Figure 12-3 Linear-quadratic differential game optimal control actions. 12-16
Figure 12-4 Three-dimensional target interception 12-17
Figure 12-5 Pursuit-evasion game trajectories 12-19
Figure 12-6 Geometry of proportional navigation 12-20
Figure 12-7 Generic seeker block diagram 12-20

Table 12-1 Information available for a discrete game 12-2

Chapter 13 - Robustness and Sensitivity

Figure 13-1 Basic control system block diagram. 13-2
Figure 13-2 A robust feedback tracking control system. 13-9
Figure 13-3 Open- and closed-loop control systems. 13-12

Chapter 14 - Precision Guided Munitions

Figure 14-1 Functional block diagram for a generic guided missile system 14-2
Figure 14-2 Operational block diagram of an active sensor 14-5
Figure 14-3 Attenuation by atmospheric gases, rain, and fog 14-6
Figure 14-4 Operational block diagram of a passive sensor 14-7
Figure 14-5 Classes of PGMs 14-9
Figure 14-6 Types of smart munitions 14-10
Figure 14-7 Generic command guidance (surface-to-air case) 14-13
Figure 14-8 Semiautomatic CLOS for an antitank missile 14-15
Figure 14-9 Homing guidance 14-16
Figure 14-10 Generic ATR components 14-20

Table 14-1 Basic homing guidance laws 14-17

Chapter 15 - Applications of Control Theory to PGMS

Figure 15-1 Block diagram of a speed control system 15-5
Figure 15-2 General air-to-air missile configuration 15-27
Figure 15-3 Missile subsystem functional block diagram 15-28
Figure 15-4 Decompositions of the problem 15-33
Figure 15-5 Schematic diagram for the implementation of proportional bang-bang guidance (PGB) 15-45

Chapter 16 - Gun Fire Control

Figure 16-1	Fire control system and target movement	16-1
Figure 16-2	Fire control system error sources	16-2
Figure 16-3	Fire control system processes	16-4
Figure 16-4	Manual track and lead system	16-6
Figure 16-5	Disturbed reticle fire control system	16-7
Figure 16-6	Stabilized sight-director fire control systems	16-8
Figure 16-7	Conventional design of estimators and predictors	16-10
Figure 16-8	3D maneuvering target in fixed 2D maneuver plane	16-12
Figure 16-9	Generalized description of 3D non linear CAAM prediction FCS (Alternate Method)	16-14

Chapter 17 - An Assessment of Modern Control Theory

Figure 17-1	Block diagram components of a guided missile	17-2
Figure 17-2	Conventional control system design	17-8
Figure 17-3	Use of an integrated design in advanced control systems	17-8

CHAPTER 1
INTRODUCTION

1.1 Assessment of Need

Demands for improved performance of tactical guided weapons expected to operate in complex air and surface battle environments necessitated the investigation and application of guidance and control systems based on modern control theory. This GACIAC State-Of-The-Art Review (SOAR) focuses on the application of selected tools and concepts drawn from modern control theory to the design and development of tactical weapon guidance and control systems.

The application of modern control theory to tactical guided weapons is a high interest, high potential technology. The basic concepts of this technology, their present application, and future potential are addressed in this SOAR. This review is intended to be of use to administrators, managers, and bench engineers, and will assist them in making informed decisions regarding future tactical weapon systems. These future systems will undoubtedly make increasing use of modern control theory technology.

Modern control theory deals with the analysis and synthesis of systems and devices for the control of complex multi-input and multi-output (MIMO) systems. As a body of knowledge, modern control theory encompasses all of classical control system design technology, augmented with a wide range of computational techniques largely developed over the past two decades.

Modern control theory is based on abstract mathematical concepts and uses a system of notation and terminology that is incomprehensible to anyone untrained in this technology. In this SOAR, we have attempted to explain the concepts, advantages, and limitations of modern control theory as applied to the guidance and control of tactical guided weapons in layman's language to the extent possible. Nevertheless, much of the mathematics has been retained to broaden the use of this report.

The essence of modern control theory is the designation of the state constants and the state variables which characterize a system. The most difficult problem is determination of the state variables. The state variables of a dynamic system are that fewest set of numbers which define the values of all variables of interest concerning a particular dynamic system or a mathematical model at a particular point in time or space. For example, a guided missile can be simply modeled by an idealized particle having a fixed mass. The motion of that particle moving through space is completely determined if the position and velocity of the particle are known at each point in time. Other variables of interest, such as the force acting on the particle or the particle's kinetic and potential energy, can be determined if we know the particle's instantaneous position and velocity and apply certain basic laws of physics. Position and velocity are thus a set of two state variables for this simple mathematical model of a guided missile.

For a dynamic system, the state variables need not be physically measurable or observable quantities. In some cases, state variables are purely mathematical entities. From a practical viewpoint, it is convenient to choose as state variables a set of variables which can be physically measured. The reason for this choice is that the modern control theory approach to control system design relies on the feedback of measured state variables to form a closed-loop control system.

Given the present availability of powerful digital computers based on miniaturized high-speed microprocessors, high-capacity random access memories, and other integrated circuits, it is now possible to include additional computational capabilities on-board newly designed tactical guided weapons, and to do so in a cost-effective manner. This capability permits the introduction and implementation of guidance and control system designs based on modern control theory as opposed to systems based on conventional classical control system design procedures.

1.2 Report Structure and Content

Despite the apparent complexity of the subject matter, modern control theory provides a unified mathematical approach to solving a wide variety of system analysis, design, and optimization problems. This GACIAC SOAR focuses on the application of selected tools drawn from modern control theory to the design and development of tactical weapon guidance and control systems.

Before moving on to details, a preview of the tools of control theory selected for inclusion in this report is in order. A summary description of the tools will also provide an outline of the chapters contained in this report.

Classical Control Theory. Classical control system design traditionally deals with single-input, single-output linear systems. Performance specifications for these systems have classically been specified in either the time or frequency domain by measures such as response or settling time, percent overshoot, bandwidth, or gain and phase margins. In a classical control system any unwanted interactions between system variables are either ignored or assumed to be minimal.

Chapter 2 presents some of the methods of classical control theory. The primary approach is that of feedback systems modeled with Laplace transforms and z-transforms for continuous-time and discrete-time systems, respectively. Both open-loop and closed-loop systems are discussed. The classical methods of stability based upon Routh's stability criterion, the root-locus method, Bode plots, and Nyquist's stability criterion and polar plots are introduced.

Modern Control Theory. In contrast to classical control theory, modern control theory allows the designer to deal with dynamic systems having simultaneous multiple inputs and outputs, thus retaining any natural interactions occurring within the total system. Additionally, modern control theory relies on the now considerable body of knowledge regarding mathematical optimization techniques as a means for designing the best possible control system.

In a design approach based on modern control theory, the desired performance of a guided weapon is specified in the language of mathematics, for example, to minimize intercept time, miss distance, or energy expended. On board a guided missile a microprocessor-based digital computer may now contain a mathematical description of the system aerodynamics and engagement kinematics in state variable form. These dynamic equations can be efficiently and rapidly processed to yield an optimal missile trajectory and a corresponding set of control surface displacements. Modern control theory can, in principle, provide the autopilot commands required to intercept the target in minimum time, or at the expense of a minimum amount of fuel, or while optimizing any other suitable function designated as the performance measure by the system's designer.

Chapter 3 provides an introduction to modern control theory and the other topics which the rest of this review addresses. The main focus in this chapter is on system modeling and the identification of state variables.

Dynamic Systems. The concept of a dynamic system is central to the application of modern control theory. A dynamic system is a physical system or mathematical model whose behavior evolves over time. A guided missile in flight is a dynamic system. The missile's position and velocity evolve over time in response to the action of its guidance, control, and propulsion systems.

Dynamic systems are discussed in Chapter 4. It is usually a routine matter to identify the inputs to a dynamic system of interest and the important outputs. For example, one input to a guided missile model is the applied thrust. The resultant outputs are the missile position and velocity.

System Identification. System identification involves the process of building a mathematical model of a dynamic system based on measurements of the system's inputs and outputs. System identification requires the determination of the structure of the mathematical model and, for a dynamic system in which the outcome does not depend on any chance factors, the evaluation of the parameters of that model.

Most design approaches in both modern and classical control theory assume that an explicit mathematical model is available for the dynamic system to be controlled. For the idealized particle model, Newton's law can be used to derive a mathematical model for the particle's acceleration as a function of the applied force and the particle's mass. The input to the resultant system is the applied force, and the eventual output is the particle's position. The structure of the mathematical model of this dynamic is thus determined.

In other methods both the structure of the mathematical model and the values of its parameters may be unknown. It is then necessary to identify both the structure of the mathematical model and the values of the model's parameters before the design of a control system can be accomplished. If, for example, the particle's mass is unknown, this unknown parameter can be experimentally identified by processing the applied input and the observed output. The mathematical parameters of a dynamic system are presented in Chapter 5.

Kalman Filter. The values of the state variables of a dynamic system must be available in order for the methods of modern control theory to compute a feedback control signal. Those state variables which can be directly measured may be used immediately in this computation. Those state variables which are not directly measurable must be estimated prior to their use in the computation. For certain classes of dynamic systems the estimation of the system's state variables can be done by means of a Kalman filter. Chapter 6 describes the structure of this important estimator.

Estimation is the assignment of a value to a variable or a coefficient in a mathematical relationship. Estimation problems differ from system identification problems. In an estimation problem the structure of the dynamic system's mathematical model is known and taken as a given. In modern control theory, estimation refers to the process of determining a specific parameter value, the values of the dynamic system's state variables, or the nature of a specific signal based on noise-corrupted measurements of the system's outputs.

As an example, suppose that a radar is to be used to measure the position of a guided missile traveling through space. The radar provides a measured value of the missile's range at a specific azimuth and elevation. The angular coordinates are available from the antenna position. Since the missile may be anywhere within the radar's effective beamwidth, its precise position is not accurately measurable by the radar, and must be estimated. This position estimate can be improved by statistically processing repeated measurements.

Adaptive Control. Adaptive control of a dynamic system, covered in Chapter 7, involves the sensing of one or more system variables and the use of that sensed data to vary feedback control systems in order to meet performance criteria. There are several related and complementary techniques which comprise the technology of adaptive control including gain scheduling, model reference adaptive control, self-tuning regulators, and designs based on optimal stochastic control theory.

The basic idea of gain scheduling is to compensate for system parameter variations by changing the parameters as a function of some auxiliary variable. This technique is commonly used to vary missile autopilot gains as a function of altitude, mach number, dynamic pressure, or some other auxiliary variable which is easily measured.

In a model reference adaptive control system, the performance specifications are given by a reference model, a mathematical description of the ideal behavior of the dynamic system. The model reference control system consists of two separate loops, an inner classical feedback loop consisting of the dynamic system being controlled and a controller, and an outer loop which alters the controller structure in response to changes in the system parameters.

The self-tuning regulator also consists of two control loops. The inner control loop consists of the dynamic system and a classical controller. The outer loop consists of a system identifier and a design calculation which yields the necessary controller structure. The self-tuning regulator directly automates the process of dynamic system modeling and control system design.

Adaptive control systems can also be developed based on stochastic control theory. The system and its environment are modeled stochastically. The performance measure minimizes the expected value of a loss function. The controller consists of a state variable estimator and a feedback signal generator. The feedback signal generator is a nonlinear device which computes the control signals based on the estimates and the input command signal.

Mathematical Optimization. Decision problems involving solutions to such problems as the best numerical values to be assigned to the parameters of a control system design, the best trajectory to be followed by a missile en route to its target, or the best input signal to apply to a dynamic system in order to drive the state variables to some desired values are all problems of mathematical optimization. Mathematical optimization plays a key role as an important tool of modern control theory. Mathematical optimization problems can be classified in many ways, some of which will be detailed later. For the moment it will be helpful to briefly mention two specific classes: static and dynamic optimization problems.

Static mathematical optimization problems involve finding the maximum or minimum value of a mathematical function of a set of variables. Each variable represents one component of a decision or potential solution to the problem. The formulation and solution of a static optimization problem do not depend explicitly on the passage of time. The variables are usually restricted by a set of constraints which limit each variable's range of values. The solution to the optimization problem requires the values of all of the variables to be selected or specified by the decision-maker. One of the principle tools for solving static optimization problems is a procedure, or algorithm, called linear programming.

While static optimization problems are generally concerned with finding a solution to a decision problem which does not involve the passage of time, dynamic optimization problems are concerned with mathematical optimization problems in which time is a factor. Dynamic optimization involves finding the solution to a mathematical problem in which the answer is a function of time rather than a set of numerical values as in a static optimization problem. The best function yields the minimum or maximum of a performance measure which is usually the value of an integral involving the initially unknown function. A set of constraints may also be operative. These constraints serve to limit or restrict the nature of the optimal solution.

The theoretical basis of dynamic optimization is a branch of applied mathematics called the calculus of variations. Many guidance and control design problems, including the development of minimum-time and minimum-energy trajectories, can be formulated as dynamic optimization problems. A principle mathematical tool for the analysis and solution of dynamic optimization problems is the algorithm known as dynamic programming. Mathematics optimization is treated in Chapter 8.

Optimal Control. The processes of mathematical modeling, state variable analysis, and system identification yield a model of a dynamic system in the form of a set of state-transition equations. These equations represent the manner in which the state variables and system output evolve over time as functions of the applied input signals. For a dynamic system which operates on a continuous-time basis, these state-transition equations will be a set of first-order differential equations, and the input and output of the dynamic system will be defined as functions of time. For a dynamic system which operates on a discrete-time basis, these state-transition equations will be a set of first-order difference equations, and the input and output of the dynamic system will be defined as sequences of numerical values.

To obtain control over any dynamic system it is necessary to introduce and apply a control input into the system and its mathematical model. Optimal control, the subject of Chapter 9, involves the selection of a particular control input for a dynamic system. The selected control function or sequence normally optimizes a performance measure which is a function of the state variables, the control input, the final system state, and the time required to reach that state. The particular performance measure to use is selected by the control system designer to reflect the overall design goals and desired system performance.

Singular Perturbation Methods. Some dynamic systems are characterized by states that may be slow or fast. It is necessary to separate a singularly perturbed dynamic system into two unique subsystems that develop based upon separate time scales. Chapter 10 analyzes such systems and describes an example for achieving optimal control.

Stochastic Control. Stochastic control theory involves problems of signal filtering, system identification, and optimal control of dynamic systems represented by noise driven differential or difference equations. The applied control action for such a system must be a function of the available information. This information often takes the form of a set of noise corrupted observations of the system state variables. Stochastic control theory as a whole is a broad subject area which also includes certain aspects of operations research including dynamic resource allocation, repair and replacement problems, and optimization problems involving finite Markov chain structures, all of which are introduced in Chapter 11.

The Markov chain structure is a basic device used to formulate a wide class of stochastic optimal control problems. A Markov chain is specified by a set of discrete states, each represented by the value of one or more state variables. Over time the dynamic system state changes in a random manner according to a set of state-transition probabilities. These probabilities are controlled by one or more control inputs. The output of the system is a function of the present state and control input. This mechanism is very similar to the state-transition mechanism commonly associated with sequential logic circuits. The objective in a stochastic control problem is to determine a control policy which minimizes or maximizes a probabilistic performance measure. The control policy is defined as a function of the observed or measured state variables. The performance measure is often the expected value of a function of the present state and control inputs.

Stochastic optimal control also concerns the determination of a control policy which optimizes the performance of a continuous or discrete-time dynamic system whose operation is affected by random disturbances, noise, or chance outcomes. Some knowledge of the statistical properties of these disturbances is presumed. The control policy may be described by an open-loop function of time alone, as a closed-loop function of the state variables, or as open- or closed-loop sequences of discrete-time control inputs. The performance measure for an optimal stochastic control problem is often the minimization or maximization of an expected value.

Differential Game Theory. The theory of optimal control applies to dynamic optimization problems in which there is one source of control inputs determined by the control system designer. The theory of differential games, Chapter 12, applies to optimal control problems in which there are several sources of control inputs, all of which interact and affect the dynamic system's state. In the language of game theory these various sources are called the players, and the outcome of the game is called the payoff.

Mathematical games of pursuit and evasion are the prototype for a large class of problems which can be investigated by means of differential game theory. In a typical problem one seeks to determine how long one player, the evader, will survive before being caught by the second player, the pursuer. In some cases the evader may escape without capture by the pursuer. The payoff in this problem might be a measure of miss distance.

There are many applications of this prototype model including aerial combat, missile versus target maneuvers, maritime surveillance, strategic balance, economic theory, and social behavior. A destroyer stalking an enemy submarine serves as a practical example for a differential game. The destroyer strives to be as near to the submarine as possible at the time depth charges are dropped. The destroyer bases its maneuvers on information it has obtained regarding the present and predicted position of the submarine. The submarine strives to maximize the distance between itself and the destroyer, and may introduce false or misleading information into the game in its efforts to avoid destruction.

Robustness and Sensitivity. A major objective of the design of modern control systems is to achieve robustness. Robustness is closely related to the system sensitivity. This relationship is described in Chapter 13.

For successful operation of the closed-loop control system it is necessary that tracking occur even if the nature or structure of the dynamic system should change slightly over the time of control. The process of maintaining the system output close to the reference input, in particular when the input equals zero, is called regulation. A control system which maintains good regulation despite the occurrence of disturbance inputs or measurement errors is said to have good disturbance rejection. A control system which maintains good regulation despite the occurrence of changes in the dynamic system's parameters is said to have low sensitivity to these parameters. A control system having both good disturbance rejection and low sensitivity is said to be a robust control system.

Precision Guided Munitions. Modern precision guided munitions (PGMs) require complex guidance and control systems. Various types of PGMs are discussed in Chapter 14. Although guidance and control are often used interchangeably, there is a fundamental distinction between the roles of a tactical weapon's guidance and its control system. The control system is responsible for automatically moving the weapons's fins, control surfaces, or thrust mechanisms thus causing aerodynamic forces and moments to be exerted on the missile. These forces and moments ultimately change the orientation and direction of the weapon's motion in space. The control problem involves the design of an autopilot, or servomechanism, which will cause the weapon to perform those maneuvers required to reach its target. These maneuvers are determined by the guidance system.

The guidance system usually contains sensing, computing, directing, stabilizing, and additional servo-control components. The guidance system processes measured or estimated data produced by the sensors concerning the position of the target relative to the weapon. The guidance system recommends changes in the flight path required for the weapon to reach its target. The guidance problem involves the design of this process, commonly called a guidance law, and the specification of the measured or estimated data necessary to compute a revised trajectory.

Applications of Control Theory. Chapter 15, brings together most of the concepts discussed in this review. Chapter 15 presents several applications of modern control theory to tactical weapon guidance and control. These examples were selected from the open literature to indicate the wide applicability of modern control theory and to display the array of design tools and approaches presently available. This report is not intended to be a handbook of design formulas for modern control, nor a cookbook of recipes for optimum solutions to tactical weapon system design problems. Rather, the reader is encouraged to pursue the topics presented in more detail, and to seek out applications for this material in those systems with which they may be involved.

Gun Fire Control. The use of gun systems to defeat stationary and moving targets from either stationary or moving platforms also requires applications of modern control theory. Gun fire control systems have benefited from many of the same technologies that have advanced missile and projectile terminal homing. A discussion of these advancements for gun systems is covered in Chapter 16.

Assessment. The final chapter, Chapter 17, summarizes the state of the art of various topics of modern control theory. Some suggestions for the future direction of these topics are presented. Modern control theory is an area that will greatly advance with new computer software, modeling, and simulation capabilities.

CHAPTER 2
CLASSICAL CONTROL THEORY

2.1 Introduction

This chapter is intended to provide the reader with a brief overview of some traditional methods used for classical control system design. These methods and their applications are outlined for comparison with modern control theory methods presented in later chapters.

Classical control theory deals primarily with single-input, single-output physical systems described by linear, constant-coefficient, time-invariant differential, or difference equations. Few physical systems, including tactical guided weapons, operate in a truly linear manner, and many are time-varying due to changes in mass or structure. However, approximations of linearity and assumptions of time-invariance allow many dynamic systems to be analyzed for their performance about nominal operating points, and for relatively small changes in their parameters and signal levels.

The traditional use of transform methods and frequency domain techniques considerably simplifies the analysis of linear, constant-coefficient time-invariant dynamic systems. Laplace transforms are the major mathematical tool for the analysis of these systems operating in continuous time, and Z-transforms are the equivalent tool for the analysis of discrete-time dynamic systems.

Using a transform method, the rather complicated differential or difference equation that defines a single-input, single-output dynamic system is literally transformed from the continuous-time or discrete-time domain to the relatively simpler transform domain. In the transform domain the dynamic system is modeled by a linear algebraic equation which can be manipulated using standard algebraic methods. Since the relationship between input and output is most often of interest, the transfer function, defined as the ratio of the transform of the output signal to that of the input signal, is an important quantity in classical control system design.

2.2 System Representations

The most prevalent system structure investigated using classical control theory is the feedback control system. A single-input, single-output continuous-time feedback control system is shown in Figure 2-1. This dynamic system is considered to be totally analog in nature. Notation commonly used in classical control theory labels the analog input or reference signal as r(t) and its

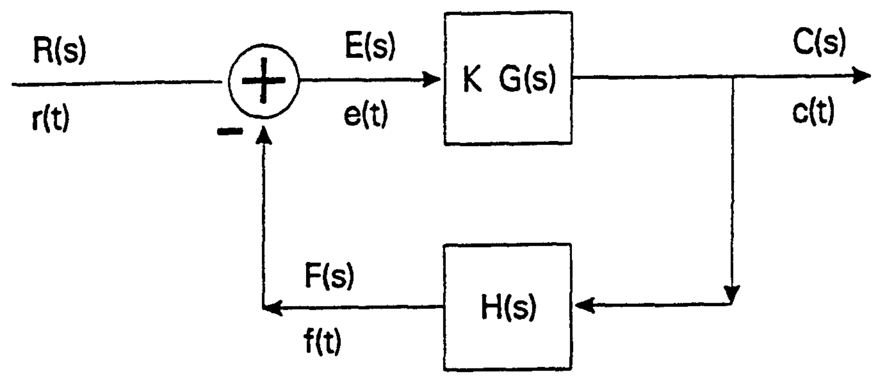


Figure 2-1. Single-input, single-output continuous-time feedback control system.

Laplace transform as R(s). The analog feedback signal is labeled f(t) and its transform is labeled F(s), the output or controlled analog signal labeled as c(t) and its transform as C(s), and the actuating or error analog signal as e(t) and its transform as E(s). To emphasize the differences in labels, x(t) refers to the mathematical function representing the value of a signal x at a time t, while X(s) represents the Laplace transform of that same signal. Tables of Laplace transforms and their corresponding time functions are readily available[2.1].

Figure 2-2 shows a feedback controller implemented for a similarly structured discrete-time dynamic system which uses a digital computer to implement the feedback, error computation, and actuation functions. It is implicitly assumed that the dynamic system operates at a constant sampling rate, and that the signal samples, provided by a set of analog-to-digital and digital-to-analog converters, are themselves separated by a sample time of T seconds. Notation commonly used in classical control theory labels the discrete-time input or reference signal as r(k) and its Z-transform as R(z), the discrete-time feedback signal as f(k) and its Z-transform as F(z), the output or controlled

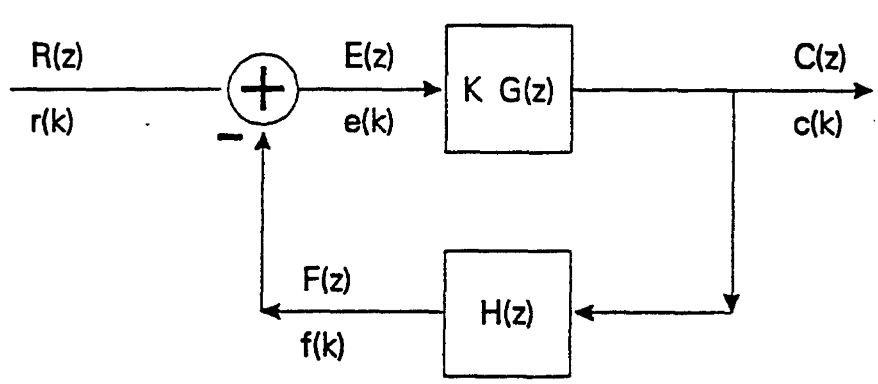


Figure 2-2. Single-input, single-output discrete-time control system.

discrete-time signal as c(k) and its Z-transform as C(z), and the actuating or error discrete-time signal as e(k) and its Z-transform as E(z). To again emphasize the difference, x(k) refers to the mathematical sequence representing the value of a signal x at a sample time indexed by k, while X(z) represents the Z-transform of that same signal. Tables of Z-transforms and their corresponding sequences are readily available[2.1].

2.3 Analysis of Continuous-Time Control Systems

In the continuous-time closed-loop control system illustrated in Figure 2-1, the forward transfer function is KG(s)—the ratio between the transforms of the output C(s) and the error signal E(s). The factor K is an adjustable gain to be selected by the designer. The factor G(s) often consists of the product of two other factors, $G(s) = G_c(s)G_p(s)$, where $G_p(s)$ is fixed and models the original dynamic system, plant, or process to be controlled and $G_c(s)$ is a compensation network or controller to be specified by the designer. The addition of $G_c(s)$ is intended to improve overall system performance.

The feedback transfer function is H(s). This transfer function represents the dynamics of the instrumentation used to form the feedback signal and any feedback signal conditioning or compensation networks. The form of H(s) is partially under the designer's control. If the feedback dynamics are sufficiently fast compared to those of the plant, H(s) is normally assumed to be a constant, and that constant is often assumed to equal unity, indicating that a direct measurement of the system output is available to form the error signal.

The open-loop transfer function is KG(s)H(s). This product of factors models the reference signal transmission through the combined plant and feedback network when the feedback signal is disconnected from the summing junction.

In the closed-loop control system illustrated in Figure 2-1, the error signal e(t) is determined by subtracting the feedback signal f(t) from the reference signal r(t). When H(s) is assigned the value of 1.0, the difference e(t) is a direct measure of the difference between the reference input r(t) and the output c(t) at the time t. Generally, the designer of a continuous-time closed-loop control system is interested in the relationships between c(t) and r(t) or e(t) and r(t), or their transform equivalents C(s), E(s) and R(s).

Using Laplace transform methods, it is quite easy to develop algebraic ratios or transfer functions involving the transforms of those signals of interest. The closed-loop transfer function between the input, R(s) and the output C(s) is:

$$\frac{C(s)}{R(s)} = \frac{KG(s)}{(1+KG(s)H(s))} .$$

The transfer function between the error, R(s) and the output, C(s), is:

$$\frac{E(s)}{R(s)} = \frac{1}{(1+KG(s)H(s))} .$$

The analog system illustrated in Figure 2-1 is a prototype for virtually all classical closed-loop continuous-time control systems. The Laplace transforms involved in these transfer functions can themselves be represented by quotients of numerator and denominator polynomials of the complex variable s:

$$G(s) = \frac{g_n(s)}{g_d(s)}$$

$$H(s) = \frac{h_n(s)}{h_d(s)} .$$

The values of s which are the roots of the denominator polynomial $g_d(s)$ are called the poles of G(s), and the values of s which are the roots of the numerator polynomial $g_n(s)$ are called the zeros of G(s). This same notion can be applied to the various transfer functions. The roots of the denominator polynomial of the open-loop transfer function KG(s)H(s) are called the open-loop poles. The roots of the numerator polynomial of the open-loop transfer function KG(s)H(s) are called the open-loop zeros.

Similarly, the roots of the denominator polynomial of the closed-loop transfer function C(s)/R(s) are called the closed-loop poles. The roots of the numerator polynomial of the closed-loop transfer function C(s)/R(s) are called the closed-loop zeros. For practical system design, numerical methods are required to factor these polynomials and determine their roots. The closed-loop system poles are especially important as they determine the system time constants, the system response to arbitrary inputs, and the relative system stability.

2.4 Analysis of Discrete-Time Control Systems

In the discrete-time closed-loop control system illustrated in Figure 2-2, the forward transfer function is KG(z), the ratio between the Z-transform of the output C(z) and that of the error signal E(z). The factor K is an adjustable gain to be selected by the designer. The factor G(z) is often the product of two other factors, $G(z) = G_c(z)G_p(z)$, where $G_p(z)$ models the dynamic system, plant, or controlled process and $G_c(z)$ is a compensation network or controller to be specified by the designer. The addition of Gc(z) is again intended to achieve an improvement in overall system performance.

The feedback transfer function is H(z), which represents the dynamics of the instrumentation used to form the feedback signal and any feedback signal conditioning or compensation networks. The form of H(z) is partially under the designer's control. If the feedback dynamics are sufficiently fast compared to those of the plant, H(z) is assumed to be a constant and the value of that constant is often taken as unity.

The open-loop transfer function is KG(z)H(z). The product of these three factors models the transmission of the reference signal through the combined plant and feedback network when the feedback signal is disconnected from the summing junction.

In the discrete-time closed-loop control system illustrated in Figure 2-2, the error signal e(k) is determined by subtracting the feedback signal f(k) from the reference signal r(k). When H(z) is assigned the value of 1.0, the difference e(k) is a direct measure of the difference between the reference r(k) input and the output c(k) at time step k. Generally, the designer of a discrete-time control system is interested in the relationships between c(k) and r(k), or e(k) and r(k), or their transform equivalents C(z), E(z) and R(z).

Using Z-transform methods, it is quite easy to develop algebraic ratios between the Z-transforms of the signals of interest:

$$\frac{C(z)}{R(z)} = \frac{KG(z)}{\left(1+KG(z)H(z)\right)} \quad \text{(closed-loop transfer function)}$$

$$\frac{E(z)}{R(z)} = \frac{1}{\left(1+KG(z)H(z)\right)} \; .$$

The closed-loop discrete-time control system illustrated in Figure 2-2 is a prototype for virtually all classical discrete-time systems of interest. Generally the various Z-transforms can themselves be represented by quotients of numerator and denominator polynomials of the complex variable z:

$$G(z) = \frac{g_n(z)}{g_d(z)}$$

$$H(z) = \frac{h_n(z)}{h_d(z)} \; .$$

The values of z which are the roots of the denominator polynomial $g_d(z)$ are called the poles of G(z), and the values of z which are the roots of the numerator polynomial $g_n(z)$ are called the zeros of G(z). This notion can be applied to the other transfer functions involved. The roots of the denominator polynomial of the open-loop transfer function KG(z)H(z) are called the open-loop poles. The roots of the numerator polynomial of the open-loop transfer function KG(z)H(z) are called the open-loop zeros.

Similarly, the roots of the denominator polynomial of the closed-loop transfer function C(z)/R(z) are called the closed-loop poles. The roots of the numerator polynomial of the closed-loop transfer function C(z)/R(z) are called the closed-loop zeros. For practical system design, numerical methods are required to factor these polynomials and determine their roots. The closed-loop system poles are especially important as they determine the time constants of the discretitive system, the response of the system to arbitrary inputs, and the relative system stability.

Much of the technical work involved in the design of control systems using classical methods involves the development of mathematical models for and the analysis of single-input, single-output feedback control systems similar to those illustrated above. Linear constant-coefficient systems having multiple inputs and outputs can be treated by extending these classical methods to dynamic systems modeled by transfer function matrices[2.1]. It is also possible to develop models for systems which are a composite of continuous- and discrete-time signals[2.2].

2.5 Reasons for Using Feedback as a Means of Obtaining Control

Continuous- and discrete-time control systems similar to those illustrated above are used to achieve the advantages of feedback control:

- the controlled dynamic system can be made to follow or track a specified input function in an automatic manner
- the performance of the closed-loop control system is less sensitive to variations in plant or process parameters
- the performance of the closed-loop control system is less sensitive to unwanted disturbances or measurement noise
- it is easier to obtain desired transient and steady-state responses

To obtain the advantages of feedback, the dynamic system must become somewhat more complicated. Unavoidable costs must be incurred and the stability of the resulting closed-loop system becomes a major design consideration.

The improper use of feedback can destabilize an otherwise stable dynamic system, while the proper use of feedback can stabilize a dynamic system previously shown to be unstable. In a feedback control system, additional gain or amplification stages may be required to compensate for signal transmission losses. While this poses no serious problem in most analog electronic or digital computer-based control systems, achieving high gain may be a serious problem in a mechanical or non-electronic control system design.

To provide the necessary feedback signals and compensation networks, additional sensors, signal summing devices, and other high-precision components are required.

2.6 Classical Closed-Loop Control System Performance Measures

A well-designed closed-loop control system should, in a classical sense, possess four desirable characteristics:

- stability
- steady-state accuracy
- satisfactory transient response
- satisfactory frequency response

These performance characteristics are discussed in more detail in the following sections.

2.6.1 Stability

A dynamic system's stability is determined by the system's response to external input signals or disturbances. An intuitive definition of a stable system is one which will remain at rest until it is excited by an external source, or one which will return to rest if all external sources are removed.

In terms of a mathematical model, stability means that the response c(t) or c(k) must not grow without bound due to a bounded input signal, an initial condition present in the system, or an unwanted disturbance. For the linear constant-coefficient time-invariant systems treated by classical control theory, stability of the closed-loop system mathematically depends only on the roots of the characteristic equation. The characteristic equation is the denominator polynomial of the closed-loop transfer function.

To ensure stability of a continuous-time system, the roots of the characteristic equation must lie in the left-half complex plane, where they have negative real parts. A negative real part corresponds to an exponentially decaying impulse response component. For a discrete-time system, the roots of the characteristic equation must lie inside the unit circle in the complex plane. A root lying inside the unit circle corresponds to a decaying sequence as an impulse response component.

There are several classical methods for determining the stability of continuous- or discrete-time systems, including Routh's criterion, the root-locus method, the use of Bode plots, the use of Polar plots and Nyquist's stability criterion, and the use of log-magnitude versus angle plots. The reader is encouraged to consult Ogata[2.1] and Dorf[2.3] for details of these methods and examples of their applications. A brief discussion of these methods is presented below.

Routh's Stability Criterion. When applied to a dynamic system modeled by a continuous-time, constant-coefficient linear differential equation, Routh's stability criterion[2.1] tells a designer whether or not there are any roots of the characteristic equation which lie in the unstable region of the complex plane. The actual locations of the roots in the complex plane are neither found nor required to be known to determine the system's stability. It is not required to factor the characteristic polynomial to apply Routh's stability criterion. This is one of the criterion's main advantages. This criterion applies to characteristic polynomials with a finite number of terms.

When Routh's stability criterion is applied to a continuous-time linear closed-loop control system, information about the absolute stability of the dynamic system can be obtained directly from the characteristic equation. If the characteristic equation is available in factored form, stability can immediately be determined by inspection of the root locations and the use of Routh's stability criterion is not required.

The procedure for Routh's stability criterion is relatively simple:

(1) Write the characteristic polynomial as

$$D(s) = a_0 s^n + a_1 s^{n-1} + \ldots + a_{n-1} s^1 + a_n = 0 \ ,$$

where all the coefficients a_i are real-valued, and a_n is not equal to zero.

(2) If any coefficient a_i is negative or zero and at least one coefficient a_j is positive, then there are one or more roots which lie in the right-hand complex plane or on the imaginary axis. Such a system is unstable.

(3) If all the coefficients a_i are positive, then arrange the coefficients of the polynomial in rows and columns as in the following:

$$
\begin{array}{lccccc}
s^n & a_0 & a_2 & a_4 & a_6 & \ldots \\
s^{n-1} & a_1 & a_3 & a_5 & a_7 & \ldots \\
s^{n-2} & b_1 & b_2 & b_3 & b_4 & \ldots \\
s^{n-3} & c_1 & c_2 & c_3 & c_4 & \ldots
\end{array}
$$

...

s^2 e_1 e_2

s^1 f_1

s^0 g_1

The coefficients b_1 , b_2, ..., are evaluated following the pattern

$b_1 = (a_1 a_2 - a_0 a_3)/a_1$

$b_2 = (a_1 a_4 - a_0 a_5)/a_1$,

$b_3 = (a_1 a_6 - a_0 a_7)/a_1$,

...

until the remaining coefficients are all zero. This pattern of cross-multiplication, subtraction, and division is repeated until the table is filled down to the row labeled s^0.

Routh's stability criterion states that the number of roots of the characteristic equation which lie in the unstable region of the complex plane is equal to the number of sign changes in the first column of the table. The absolute stability of a continuous-time dynamic system described by a linear, time-invariant, constant-coefficient differential equation is thus simply determined by the application of Routh's stability criterion.

To apply Routh's stability criterion to a discrete-time system, the bilinear transformation $z = (w+1)/(w-1)$ is used to map the inside of the unit circle in the complex z-plane into the left half of the complex w-plane. The use of this algebraic transformation converts the system's characteristic equation into a polynomial in w, to which the designer applies Routh's criterion in exactly the same manner as for a continuous-time system.

The Root-Locus Method. The stability and transient response of a closed-loop system is determined by the location of the poles of the closed-loop transfer function. In the classical analysis of single-input, single-output control systems, it is necessary to determine the location of the poles in the complex plane. When designing a closed-loop control system, the location of the open-loop poles and zeros are adjusted by the designer so as to place the resulting closed-loop poles and zeros at desirable locations in the complex plane.

The closed-loop poles are the roots of the characteristic equation. Finding the locations of these poles, in general, requires factoring the characteristic polynomial. This is classically a tedious task if the degree of the characteristic polynomial is greater than two. Classical algebraic techniques for factoring polynomials are not well-suited for use in this application because as the designer changes the gain or any other system parameter, the location of the closed-loop poles changes and the computations must be repeated.

W. R. Evans developed a simple graphical method for finding the roots of the characteristic equation, and this method, called the root-locus method[2.1], is now used in classical control system design. In the root-locus method, the roots of the characteristic equation are plotted for all possible values of a single system parameter such as gain. The root locations corresponding to one particular numerical value of this parameter can then be determined by inspection of this plot, or root-locus. The parameter of interest is usually the open-loop gain but the influence of any other parameter of interest can be investigated.

Since the characteristic equation for a continuous-time dynamic system is given by $1 + KG(s)H(s) = 0$, the values of s which satisfy the characteristic equation must be those which make the product $KG(s)H(s)$ equal to -1. Evan's root-locus method enables a designer to determine the locations of the closed-loop poles from an analysis of the open-loop transfer function's poles and zeros with the gain K as a parameter. The method provides an indication of the way in which the open-loop transfer function must be modified so that the resulting closed-loop system is stable and meets the performance specifications.

For a discrete-time dynamic system the characteristic equation is $1 + KG(z)H(z) = 0$. The stability region for a discrete-time system is the inside of the unit circle. Application of the root-locus method is essentially the same for either continuous- or discrete-time systems.

Bode Plots. A Bode plot is a graphical method which provides stability information for minimum phase systems—systems which have no open-loop poles or zeros in the unstable region of the complex plane. A Bode plot is a logarithmic plot of the magnitude and phase angle of the open-loop transfer function versus frequency. For a continuous-time system, the sinusoidal transfer function, or frequency response, can be obtained by the substitution $s = j\omega$, where $\omega = 2\pi f$ is the angular frequency in radians per second and f is the frequency in Hertz. The product $KG(j\omega)H(j\omega)$ is then plotted in terms of its magnitude and phase angle versus radian frequency ω on two separate plots.

The classical Bode plot method is well-suited for graphical analysis if the open-loop transfer function is available in factored form, since straight-line asymptotic approximations can be used for each factor. The critical point for stability on a Bode plot is that frequency ω at which the magnitude of the open-loop transfer function equals 1.0, or 0 dB, and the phase angle of the open-loop transfer function equals $-180°$.

Bode plot methods can be applied to discrete-time systems by first applying the bilinear transformation $z = (w+1)/(w-1)$ to map the inside of the unit circle in the complex z-plane into the left half of the complex w-plane, and then substituting $w = j\omega'$. When this process is applied, the transformed frequency ω' is a distorted representation of the true sinusoidal frequency. For this reason, Bode plot methods, as well as the classical Nyquist and log-magnitude methods described below, are not often applied in the classical design of discrete-time control systems.

<u>Nyquist's Stability Criterion and Polar Plots</u>. The characteristic equation for a continuous-time dynamic system is given by $1 + KG(s)H(s) = 0$, where the complex variable s can be written as the sum of a real and an imaginary part, $s = \sigma + j\omega$. In a polar plot the product $KG(j\omega)H(j\omega)$ is plotted as a complex vector having a magnitude and phase angle with the frequency ω as a parameter. The critical point for stability on this plot is the point -1, where the magnitude is unity and the phase angle is $-180°$.

Nyquist's stability criterion, which applies to all systems whether or not they are minimum phase systems, states that the number of unstable closed-loop poles is $Z = P - N$, where P is the number of open-loop poles in the right half of the complex plane and N is the number of encirclements of the critical point made by the polar plot. Counterclockwise encirclements are taken to be positive when applying this method. A minimum phase system is a linear, constant-coefficient dynamic system whose transfer function has no open-loop poles or zeros in the unstable region of the complex plane.

<u>Log Magnitude Versus Phase Angle Plots</u>. These plots contain the same information as a Bode plot, but the magnitude and phase angle are combined on a single plot with the radian frequency ω as a parameter.

2.6.2 Steady-State Accuracy

A controlled dynamic system which has satisfactory steady-state accuracy is one in which the error signal, e(t) or e(k), rapidly approaches zero or a sufficiently small value as time increases. The Laplace-transform final value theorem is classically used to analyze this requirement without the need to actually solve for the response of the dynamic system to any test input. For a continuous-time system:

$$\lim t \rightarrow \infty \; e(t) = \lim s \rightarrow 0 \; [sE(s)] ,$$

while for discrete-time systems, the corresponding Z-transform final value theorem is:

$$\lim_{k \to \infty} e(k) = \lim_{z \to 1} \left[(z-1)E(z)\right] .$$

The results obtained by this process are valid when the indicated limits exist. A set of test input signals, the step, ramp and parabola, are assumed, and a set of static error coefficients called the position, velocity, and acceleration coefficients are then developed. The values of these coefficients provide a measure of the system's ability to closely follow both the test input signal and other arbitrary inputs.

For practical applications, Table 2-1 from Dorf[2.3] can be used. In this table a continuous-time dynamic system is characterized by the parameter Type, the number of integrations existing in the forward transfer function. For Type 0, 1, and 2 dynamic systems Table 2-1 indicates the static error coefficients for unit step, ramp, and parabolic input signals.

TABLE 2-1. SUMMARY OF STEADY-STATE ERRORS

NUMBER OF INTEGRATIONS IN G(s), type number	INPUT		
	STEP, r(t) = A R(s) = A/s	Ramp, At, A/s^2	Parabola, $At^2/2$, A/s^3
0	$e_{ss} = \frac{A}{1 + K_p}$	Infinite	Infinite
1	$e_{ss} = 0$	$\frac{A}{K_v}$	Infinite
2	$e_{ss} = 0$	0	$\frac{A}{K_a}$

2.6.3 Satisfactory Transient Response

A controlled dynamic system having a satisfactory transient response is one in which, for abrupt changes in the input or reference signal, there is no excessive overshoot, an acceptably small amount of oscillation at an acceptable frequency, an appropriate final value, and a satisfactory speed of response or settling time.

All of these transient response factors are interrelated. All depend on the location of the closed-loop poles in the complex plane, and the closeness of these poles to the appropriate stability boundary. The term, "relative stability," is sometimes used to describe the performance of a stable dynamic system in response to test inputs.

A root-locus plot indicates directly the location of the closed-loop poles of a proposed system, and in classical control system design the root-locus plot is most often used to study transient response issues for either continuous-time or discrete-time systems. Bode plots, Nyquist plots, and log-magnitude versus phase angle plots can only give indirect information regarding the transient response of a system, and, as a result, these methods are more suited to the investigation of frequency response questions.

Two classical measures used to indirectly provide a measure of a system's stability and transient response are the gain margin (GM) and the phase margin (PM). The GM is the additional gain which may be inserted in a system, with no change in phase angle, and still maintain a stable system. On a Bode plot the GM is measured by the vertical distance from the open-loop magnitude curve to the 0 dB line at the frequency where the indicated phase angle is $-180°$. The GM will be positive if the open-loop magnitude curve is above the 0 dB line at the frequency where the indicated phase angle is $-180°$.

The PM is the additional phase shift which may be inserted in a system, with no change in gain, and still maintain a stable system. On a Bode plot this is the vertical distance from the open-loop phase angle curve to the $-180°$ line at the frequency where the indicated magnitude is 0 dB. The PM will be positive if the open-loop frequency curve is above the $-180°$ line at the frequency where the indicated magnitude is 0 dB.

Classical design practice based on experience dictates that a system having an acceptable transient response will have gain and PMs of about:

$$\text{PM} > 30°$$

$$\text{GM} > 6 \text{ dB} .$$

Many control systems have their transient response dominated by a pair of complex poles lying in the left-hand complex plane. Analytical results for second-order systems can be used in this case, and the following rules of thumb can be applied:

$$\text{Damping ratio} \approx 0.01 \text{ PM (PM in degrees)}$$

$$\text{Percent overshoot} \approx 75 - \text{PM}$$

$$\text{(Rise time)} * \text{(closed-loop bandwidth, rad/sec)} \approx 0.45(2\pi) .$$

2.6.4 Satisfactory Frequency Response

A controlled dynamic system which has a satisfactory frequency response will have an acceptable bandwidth, a finite maximum gain from input to output, an acceptable frequency at which the highest gain occurs, and adequate gain and PMs. Bode plots, Nyquist plots, and log-magnitude versus phase plots are classical tools for investigating the frequency response of continuous-time and discrete-time dynamic systems by evaluating the open-loop transfer function.

To determine the closed-loop frequency response of a continuous-time system, a Nichol's chart, which performs a conversion from the open-loop frequency response to the closed-loop frequency response, is often used. When the required computations are done by hand it is common classical design practice to first use an open-loop method to develop the necessary open-loop magnitude and phase angle information. This open-loop data is then plotted in Nichol's chart format.

A digital computer can conveniently be used to calculate open- and closed-loop frequency responses in terms of complex numbers, converting the components to a magnitude, and phase angle. The magnitude may then be converted to decibel (dB) format for display and output and a computer-driven plotting routine can be used to present the resulting frequency response curves.

2.7 Classical Methods for Improving Performance

The classical analysis methods outlined above also serve as design methods. A trial and error procedure is used, in which the designer analyzes the present system's performance, decides on a modification to the system, and then re-analyzes performance to verify the design's success. The addition of feedback loops or the addition of compensation networks can be analyzed by the use of root-locus, Bode, or Nyquist methods. Any modifications made will reshape the root-locus and modify the gain and PMs of the system. The static error coefficients will also be altered.

When the performance of a single-input, single-output dynamic system is not satisfactory in terms of its frequency or transient response, stability, or steady-state accuracy, the following four classical remedies can be considered.

(1) The open-loop gain K can be adjusted. The amount of adjustment required can be estimated by one of the previous analysis methods. For example, a root-locus plot will reveal the changing location of the closed-loop poles as the gain is varied. Since the root-loci are well defined, it may be the case that no point on the root-locus will give acceptable performance. In that case the system structure must be modified by the addition of other components.

(2) It may be possible to change the structure of the system slightly, perhaps by the addition of other feedback signals. The addition of a minor feedback loop will alter the shape of the root-locus, and change the closed-loop pole locations for each value of gain K. In missile systems, the addition of rate and acceleration feedback loops is a common means of improving the stability and performance of the resulting closed-loop system.

(3) The addition of compensation networks in the forward or feedback path, or the use of discrete-time compensation algorithms, can alter the root-locus and change the magnitude and phase characteristics of the system so as to yield satisfactory performance.

(4) Major changes in the system structure may require redesign and the substitution of higher performance components. For example, achieving very high gains is not a problem in an electronic system, but the creation of high mechanical gains may require hydraulic rather than electric motors.

An alternative method of design by means of synthesis rather than repeated analysis is also available. In the synthesis approach, the required specifications and performance are translated into a desired closed-loop transfer function. For example, if the desired closed-loop transfer function is specified by M(s), it can be algebraically related to the required compensator $G_c(s)$:

$$M(s) = \frac{(G_c(s)G(s))}{(1+G_c(s)G(s)H(s))}$$

$$G_c(s) = \frac{M(s)}{((1-M(s)H(s)))} .$$

When designing an analog electronic system, this result places certain technical restrictions on the desired M(s) if the compensator $G_c(s)$ is to be physically realizable in terms of passive resistive, capacitative, and inductive components. For a discrete-time system, the variable z replaces s in the preceding equations. Since a discrete-time system is implemented in an algorithm, or computer program, there is no need for the designer of a discrete-time digital compensator to be concerned about physical realizability. For this reason, the synthesis method of design is more widely used for the design of discrete-time systems, while the analysis approach is favored for continuous-time classical design.

One additional design parameter which enters into a discrete-time system is the sample time, T. According to Nyquist's sampling theorem, an arbitrary band-limited signal must be sampled at a rate corresponding to twice the highest frequency component of the signal. This highest frequency is also a measure of the bandwidth required of the control system. In practice, a sampling rate of more than twice the highest frequency of interest is used. In a closed-loop system, the sample time T interacts with the gain K and affects the locations of the closed-loop poles and, in turn, system stability.

2.8 Classical Performance Measures and Analytical Methods

Classical control theory relies on the use of system models specified in terms of transfer functions or block diagrams to answer the following three general questions about the control of linear constant-coefficient time-invariant systems:

- What are appropriate measures of system performance that can be easily applied to develop a feedback control system?
- How can a proposed feedback control system be analyzed in terms of these performance measures?
- How should a control system designer modify a system if its performance is unsatisfactory?

The methods of analysis used in classical control theory were developed before the widespread use of computers, and, as a result, these methods strive to develop as much information as is possible about the response of a system, c(t) or c(k), to an arbitrary input r(t) or r(k), without the necessity of solving the system's dynamic equation for every possible input signal.

The problem of considering an infinite variety of input signals was solved by relying on a standard set of mathematical test inputs. Step functions, ramp functions, and sinusoids are all used to develop estimates of a classical control system's performance.

The way in which several of these classical methods can be used to design both continuous- and discrete-time control systems will next be shown by a series of examples. The examples are intentionally simplified to introduce the reader to the classical approach to closed-loop control system design.

2.9 Continuous-Time Control System Design Example

Figure 2-3 shows an open-loop system proposed to control a portion of a tactical guided missile. This dynamic system consists of an electronic power amplifier, which converts a low-voltage command input signal into a high-voltage servomotor input signal, a servomotor which responds accurately to its input signal and produces a mechanical torque output, a gear box which links the servomotor to the missile control surface, and a potentiometer which serves as a sensor measuring the control surface position.

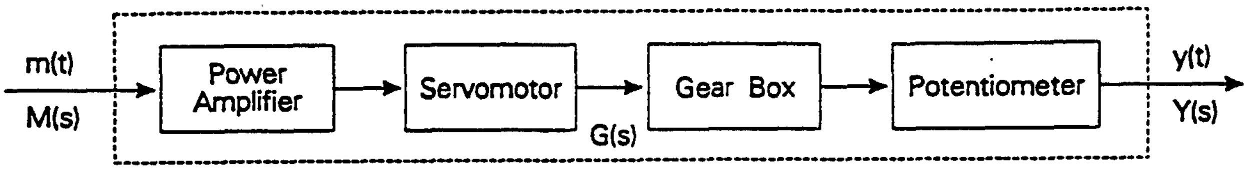


Figure 2-3. Open-loop system components.

The open-loop system has a high gain, but is clearly unstable due to the presence of a pole at s = 0. The open-loop transfer function of this dynamic system is:

$$G(s) = \frac{45000}{s(s+2)(s+30)} .$$

This transfer function was obtained by analyzing the results of a frequency response test performed on the open-loop system.

Since this transfer function is available in factored form, a Bode plot design method will be used to develop a closed-loop control system which is stable and has an acceptable level of performance. The design problem is to develop a closed-loop system which meets the following performance specifications:

- GM greater than 5 dB
- PM greater than 45°
- steady-state error in response to a unit ramp input of less than or equal to 0.05

Since the dynamic system is a type-one system having a single pole at s = 0, the transient response of the resulting closed-loop system to a unit step input will involve an initially uncertain amount of overshoot, but will eventually settle to a steady-state error of zero. The transient response to a unit ramp input will involve a constant, predictable steady-state error. The system will follow the commanded ramp input, but in the steady state, the output will never quite equal the input. These transient responses will be graphically presented later.

Based on experience, the designer elects to insert a compensation network in the forward path and to employ unity feedback. This compensation network has the transfer function G(s), and the resulting closed-loop system is shown in Figure 2-4.

Figure 2-5a is a Bode plot for the uncompensated open-loop system. Note that this plot consists of two curves—the upper, Figure 2-5a, showing the magnitude of the frequency response in decibels, and the lower, Figure 2-5b, showing the phase angle of the frequency response in degrees. Future Bode plots will not give separate titles for the two curves. The GM of the uncompensated

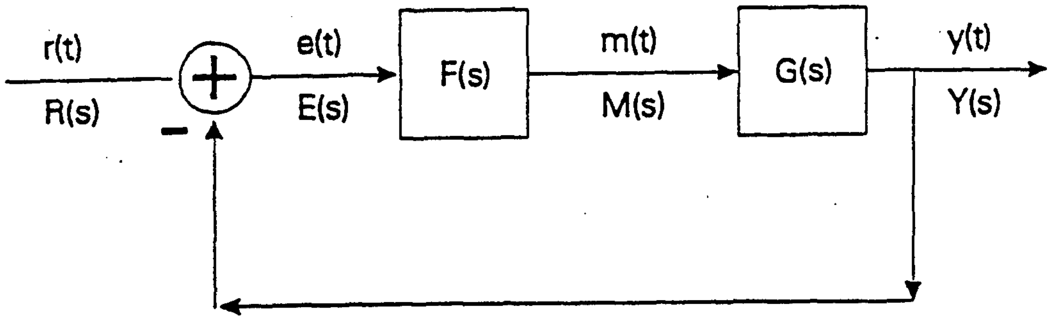


Figure 2-4. Closed-loop servomechanism.

system is about −27 dB, and the PM of the uncompensated system is about −42°. The uncompensated open-loop system is unstable. To improve the performance of this system, the phase angle which now occurs at that frequency where the open-loop gain is 0 dB must be increased, and the GM must be made positive. This can be accomplished by the use of a phase-lead network. The phase-lead network has the following transfer function:

$$G_c(s) = \frac{K(1+\tau s)}{(1+\tau\alpha s)} .$$

The parameters τ and α are selected by the designer to place the pole and zero of the compensation network at locations which result in a stable closed-loop system. A procedure for designing a phase-lead compensator has been outlined by Dorf[2-3], and will be used in this example.

The phase-lead compensation network can be designed by completing the following steps:

(1) Evaluate the uncompensated system PM when the steady-state error conditions are satisfied. The steady-state error conditions are satisfied by adjusting the gain of the uncompensated open-loop system. This adjustment may be accomplished by a hardware adjustment or by the addition of an auxiliary amplifier.

(2) Determine the necessary additional phase lead, ϕ_m, allowing for a small additional phase angle safety margin.

(3) Evaluate the parameter α:

$$\alpha = \frac{\left(1+\sin(\phi_m)\right)}{\left(1-\sin(\phi_m)\right)} .$$

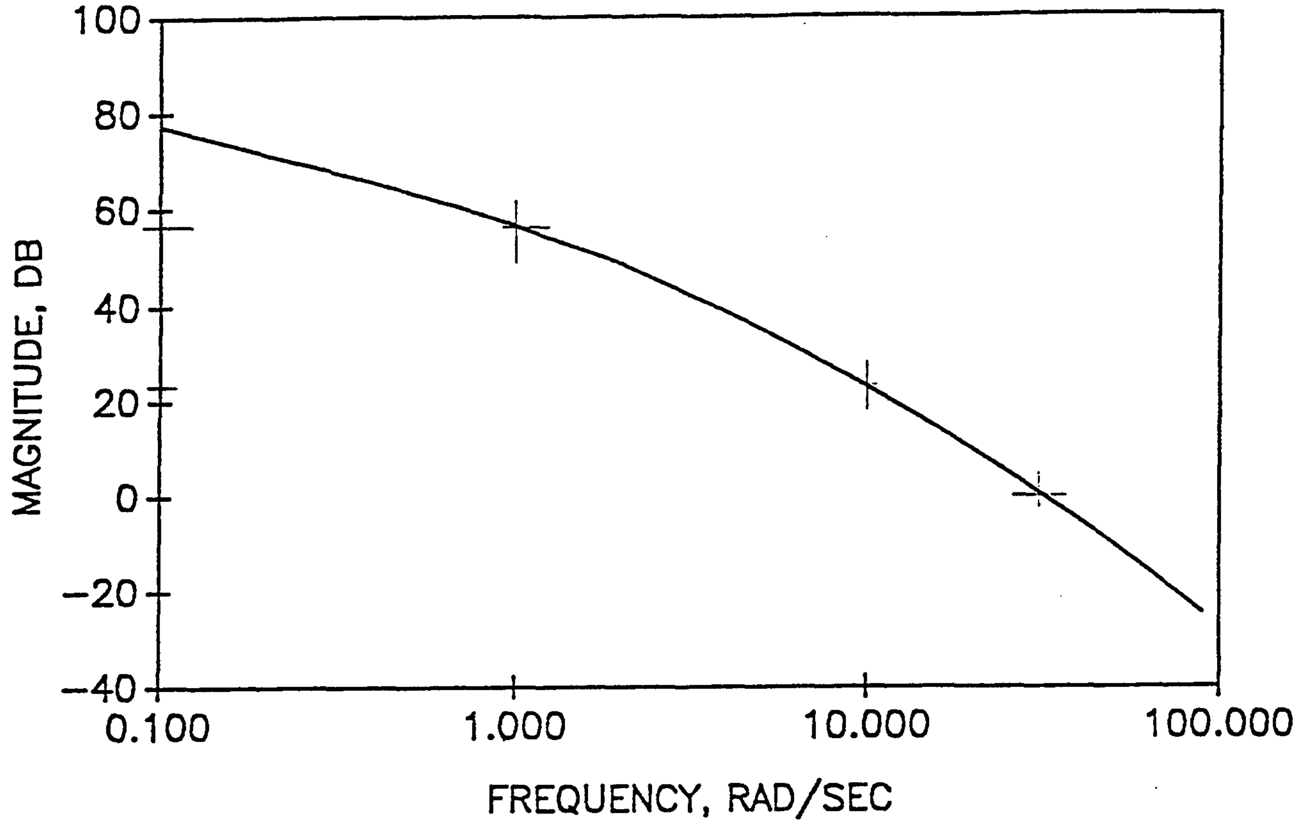


Figure 2-5a. Frequency response in decibels for a Bode plot of an uncompensated open-loop system.

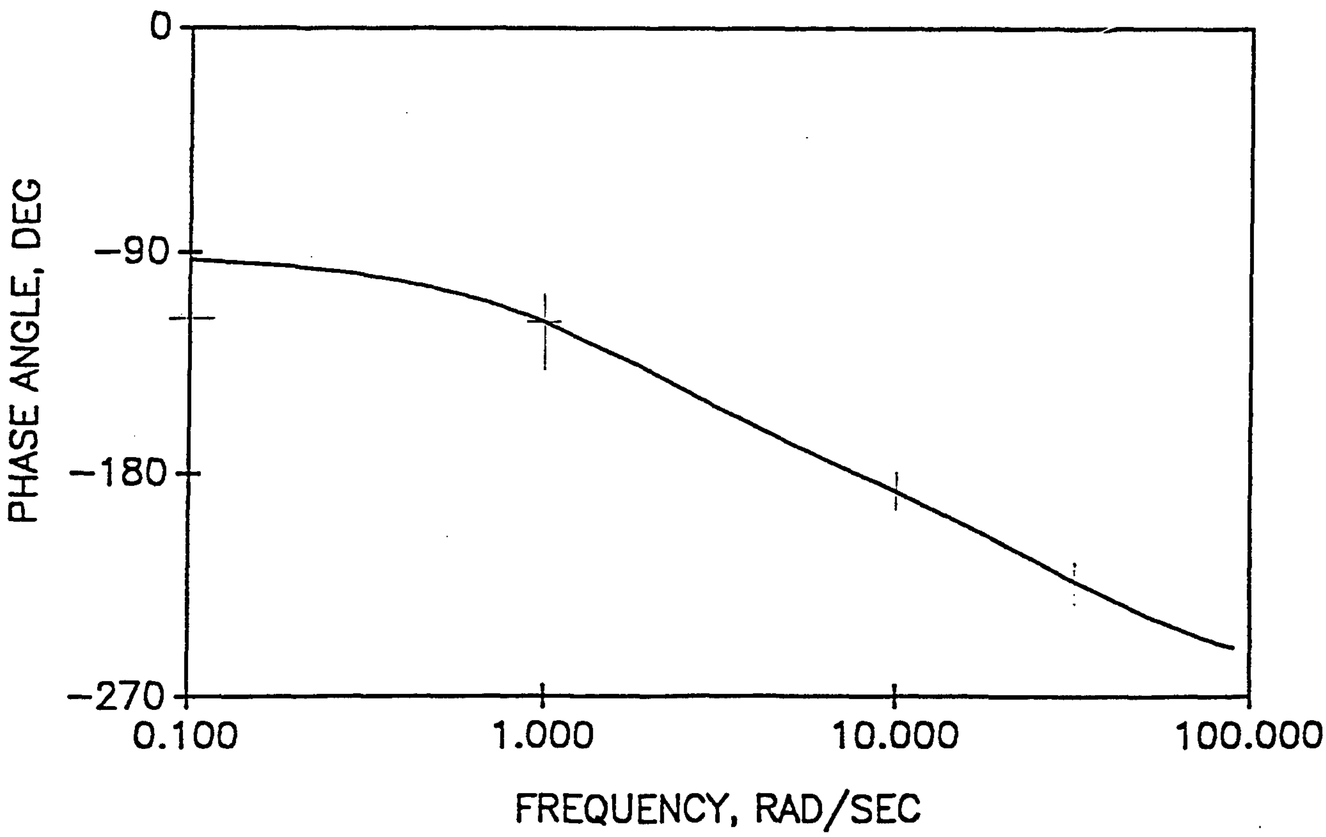


Figure 2-5b. Phase angle of the frequency response in degrees for a Bode plot of an uncompensated open-loop system.

(4) Evaluate the value 10 log(α). Determine the frequency where the uncompensated magnitude curve equals -10 log (α). This frequency will be the compensated 0 dB crossover frequency, and the compensated system bandwidth ω_m. The compensator provides a gain of 10 log(α) at the frequency ω_m. The parameter τ is determined from:

$$\tau = \frac{1.0}{\left(\omega_m \sqrt{(\alpha)}\right)} \ .$$

(5) Construct the Bode diagram for the compensated open-loop system, check the resulting gain and PMs, and repeat the design steps if necessary.

The following steps will next be executed for the design problem presented above.

__Step 1__. The steady-state error requirement of 0.05 requires that the velocity coefficient K_v equal 1.0/0.05, or 20.0. To achieve a gain change, an auxiliary amplifier having a gain of K_a is added to G(s). Then:

$$K_v = \frac{1.0}{0.05} = 20.0 = \lim s{\rightarrow}0 \left[sG(s)\right]$$

$$K_v = \lim s{\rightarrow}0 \left[\frac{s\ 45000\ k_a}{(s+2)\ (s+30)}\right]$$

$$K_a = 0.0267 = \frac{1}{37.5} \ .$$

The required gain will be obtained by adjusting the gain of G(s) plus the auxiliary amplifier to be equal to 1200. Figure 2-6 is a Bode plot of the uncompensated open-loop system when the gain is adjusted to 1200.

The PM of the uncompensated system with a gain of 1200 is about $+7°$. The GM of the uncompensated system is about $+5$ dB. The closed-loop system resulting from only a gain change is marginally stable, and would exhibit severe overshoot and sustained oscillations in response to unit step and ramp test inputs.

__Step 2__. The specified PM is 45°, and the PM resulting from Step 1 is 7°. The necessary additional phase lead is 45° $-$ 7% degrees = 38° + 10% degrees = 42°. The additional required phase lead ϕ_m is 42°.

__Step 3__. The parameter α is computed as $\alpha = (1+\sin(42°))/(1-\sin(42°)) = 5.044$. Select alpha = 5.0 for convenience.

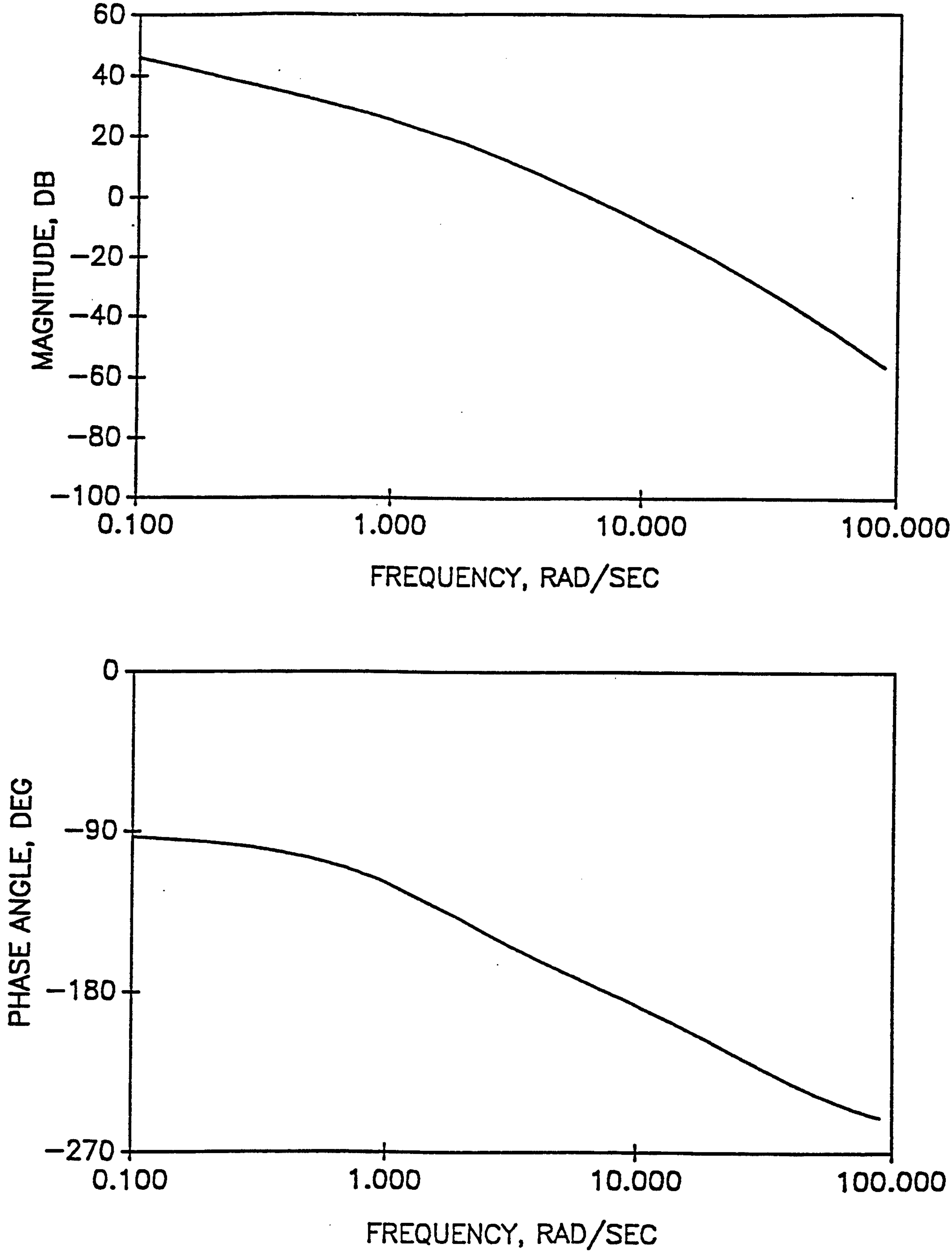


Figure 2-6. Bode plot for an uncompensated open-loop system with a gain of 1200.

Step 4. The value 10 log(α) = 10 log(5.0) = 7.02 dB. The compensated crossover frequency ω_m occurs at about 9.5 radians per second, where the uncompensated magnitude curve equals −7.02 dB. The parameter τ can now be determined:

$$\tau = \frac{1.0}{\left(\omega_m \sqrt{(\alpha)}\right)} = \frac{1.0}{\left(9.5\sqrt{(5.0)}\right)} = 0.047\ .$$

Select τ = 0.05 for convenience.

The transfer function of the compensator is:

$$G_c(s) = \frac{1}{5}\,\frac{(1+0.25s)}{(1+0.05s)}\ .$$

The gain of the auxiliary amplifier must now be increased by a factor of 5 to account for the factor 1/5 in the compensator transfer function. K_a then becomes 0.133.

Step 5. Figure 2-7 is a Bode plot for the compensated, gain-adjusted open-loop system. The resulting PM is about 37°, somewhat less than required but satisfactory, and the resulting GM is about 12 dB, somewhat better than required. These results, which do not exactly meet the design specifications, indicate the need for an iterative approach to classical control system design.

The original open-loop system was a third-order system. The addition of the compensator makes the composite system a fourth-order system. The rules of thumb applied to second-order systems can be used to estimate the damping factor, (0.01 PM), as 0.37, indicating a lightly damped system, and a percent overshoot of 25% in response to a unit step input. Figure 2-8 illustrates the transient response of the compensated closed-loop system for unit step and ramp inputs. Note that the measured percent overshoot is about 40%, and the steady-state error for a ramp input is 0.05 as required. A block diagram of the complete closed-loop control system, including the compensator, is shown in Figure 2-9.

The transient responses illustrated in Figure 2-8 were obtained by assigning a state variable to each of the integrators, assuming all initial conditions to equal zero, and numerically integrating the resulting first-order differential equations by means of a rectangular integration process with a step size of 0.005 seconds.

Several approximations were made during this design example, including the selection of convenient numerical values for the parameters α and τ. If the resulting transient performance was determined to be unsatisfactory, the designer would repeat the process, adjusting these values until the performance specifications were met as closely as possible. In some cases, it may be impossible to

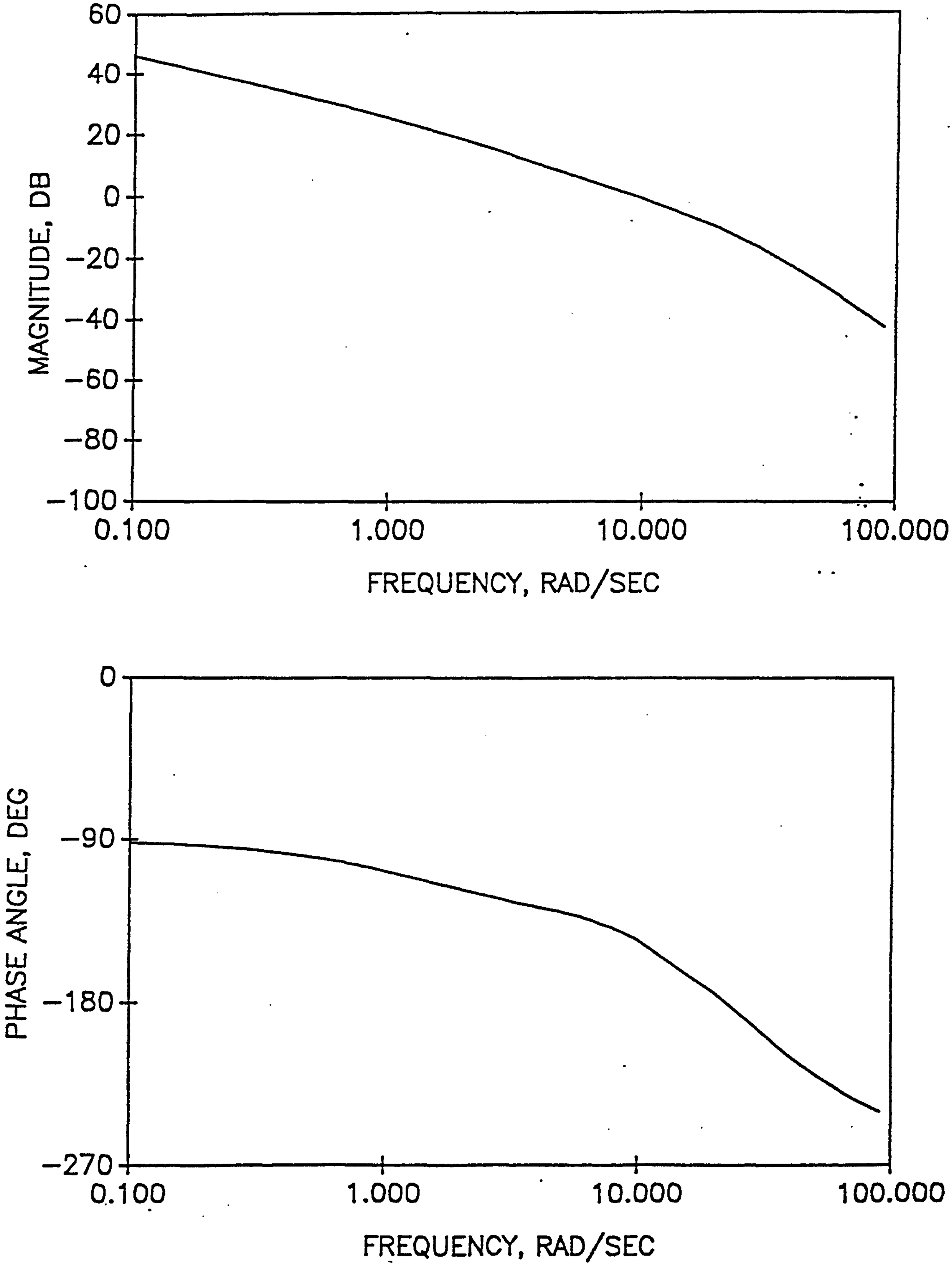


Figure 2-7. Bode plot for a compensated, gain adjusted (gain = 1200) open-loop system.

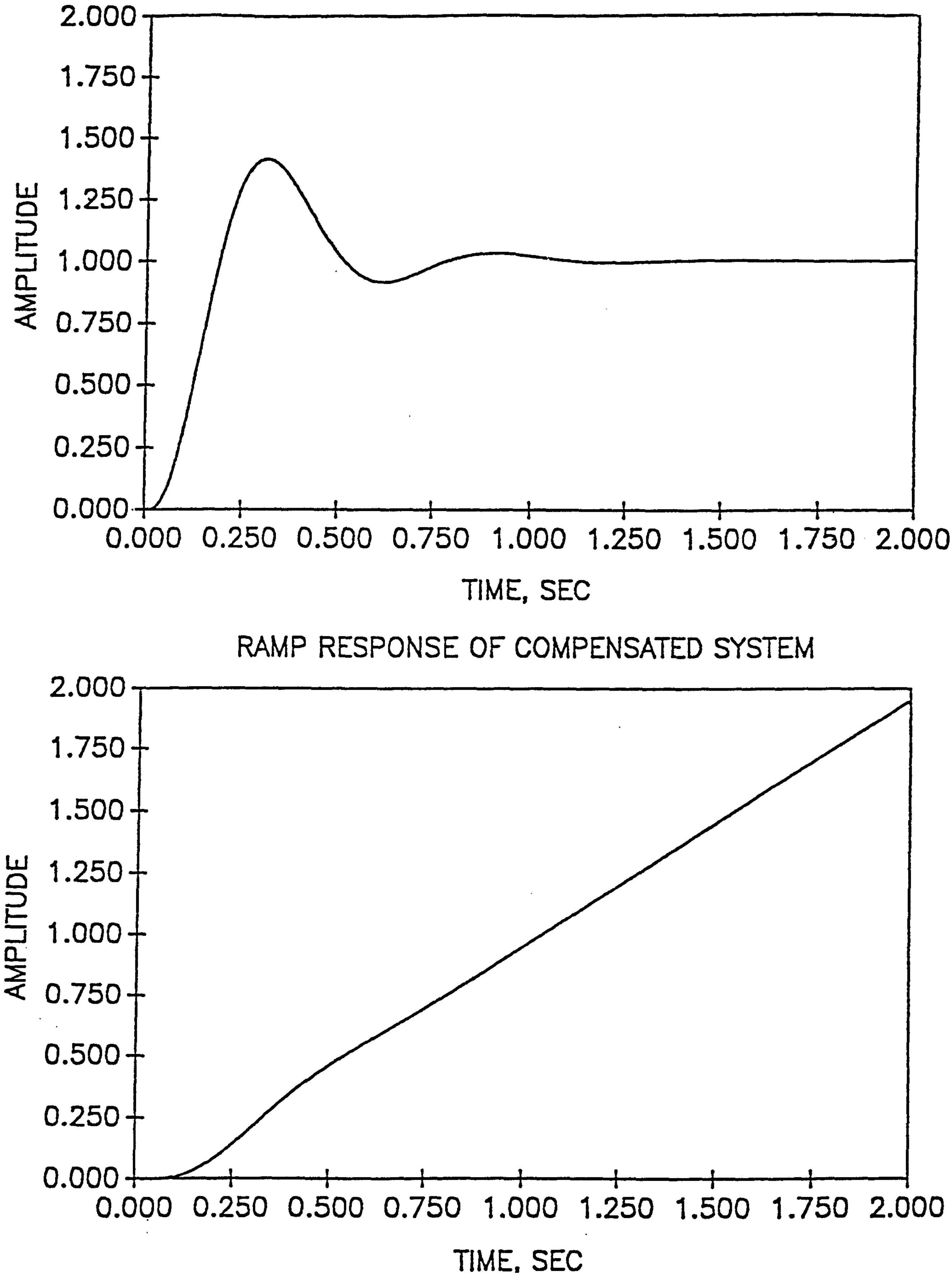


Figure 2-8. Transient response of a compensated closed-loop system for unit step and ramp inputs.

satisfy all the design requirements without substantial modifications to the structure of the underlying dynamic system.

2.10 Discrete-Time Control System Design Example

Discrete-time control systems similar to that illustrated in Figure 2-10 can also be designed by Bode plot methods. In Figure 2-10 the controlled system operates in continuous time and is described by the transfer function G(s). The discrete-time nature of the control problem results from the use of a digital computer to control the signal sampling process and to execute the algorithm corresponding to D(z), a discrete-time compensator.

The control system design problem is to devise a suitable compensation algorithm which will result in good closed-loop system behavior. The analog-to-digital (ADC) and digital-to-analog (DAC) converters shown in the figure are necessary to sample the various signals and transform them between the discrete-time digital computer domain and the continuous-time domain of the original open-loop system. The DACs and ADCs are simultaneously clocked and sampled at a time interval of T seconds. A small delay, corresponding to the time required to execute the compensation algorithm, is anticipated and, if sufficiently small relative to any resulting closed-loop system time constants, is ignored.

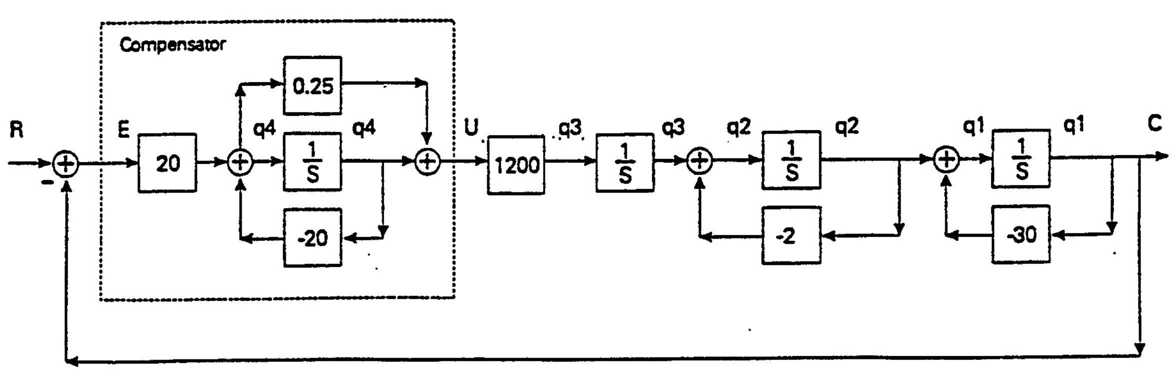


Figure 2-9. Compensated closed-loop system with a gain of 1200.

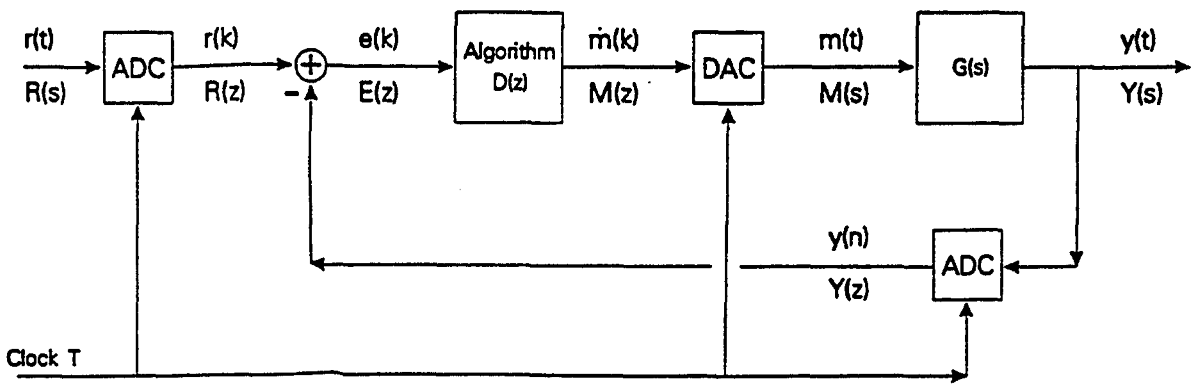


Figure 2-10. Discrete-time closed-loop control system.

The transfer function of the uncompensated open-loop system is the same as that considered in the previous continuous-time design example:

$$G(s) = \frac{45000}{s(s+2)(s+30)} .$$

If the sampling interval T is sufficiently small compared to the time constants and natural frequencies of the compensation algorithm D(z), the design of the compensation algorithm can be initiated by first designing a continuous-time compensator as in the prior example, and then determining an equivalent discrete-time algorithm by one of several means. In the prior continuous-time design example, a continuous-time compensation network was designed and implemented in analog form:

$$G_c(s) = \frac{(1+0.25s)}{37.5\,(1+0.05s)} .$$

Note that in this example the auxiliary amplifier's gain adjustment factor 1/37.5 = 0.0267 has been included in the transfer function of the compensator, and the factor 1/5 has been eliminated. The bilinear transformation can be used to develop a discrete-time equivalent transfer function. This is done by replacing the variable s by the following expression:

$$s = \frac{2\,(1-z-1)}{T\,(1+z-1)} .$$

After algebraically clearing all terms, and assuming a value for the sample time T equal to 5 milliseconds (0.005 seconds), the following discrete-time compensator transfer function is obtained:

$$D(z) = \frac{0.0267\,(101z-99)}{(21z-19)}\ .$$

A simulation block diagram for this compensator can be obtained by the use of an auxiliary variable technique:

$$D(z) = \frac{Y(z)}{X(z)} = \frac{Y(z)\,W(z)}{W(z)\,X(z)} = \frac{0.0267\,(101z-99)}{(21z-19)}\ .$$

Let $\dfrac{Y(z)}{W(z)} = 0.0267\,(101z-99)$ and

$$\frac{W(z)}{X(z)} = \frac{1}{(21z-19)}\ .$$

Then $Y(z) = 0.0267\big(101zW(z) - 99\,W(z)\big)$ and

$$zW(z) = \left[\frac{1}{21}\right]\big(X(z) + 19W(z)\big)\ .$$

Figure 2-11 shows the resulting discrete-time compensator structure and the structure of the resulting closed-loop control system. The block labeled z^{-1} represents a time delay of one sample time. The dashed line indicates the compensation algorithm performed by software imbedded in a control microprocessor.

The transient response of the discrete-time closed-loop control system is shown in Figure 2-12. Note that the sample time of this system is 5 milliseconds. If the sample time is altered, the compensation algorithm must also be changed.

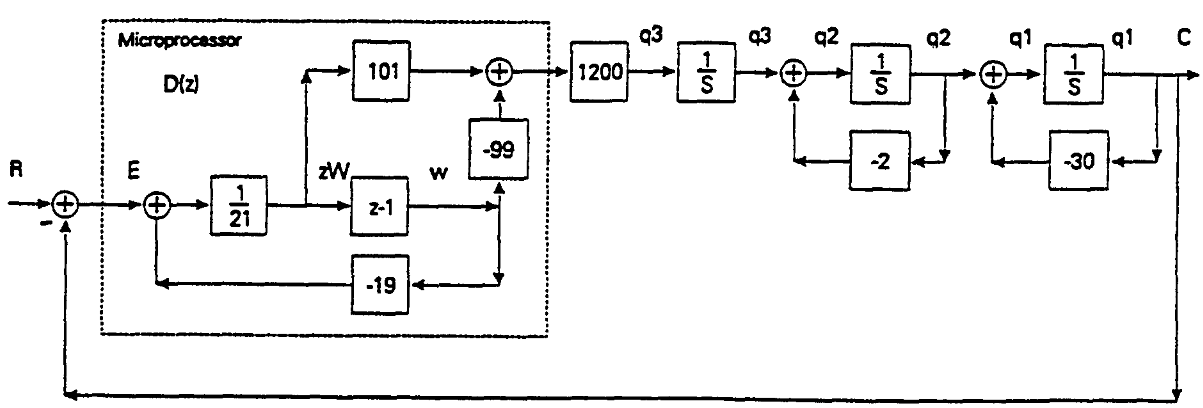
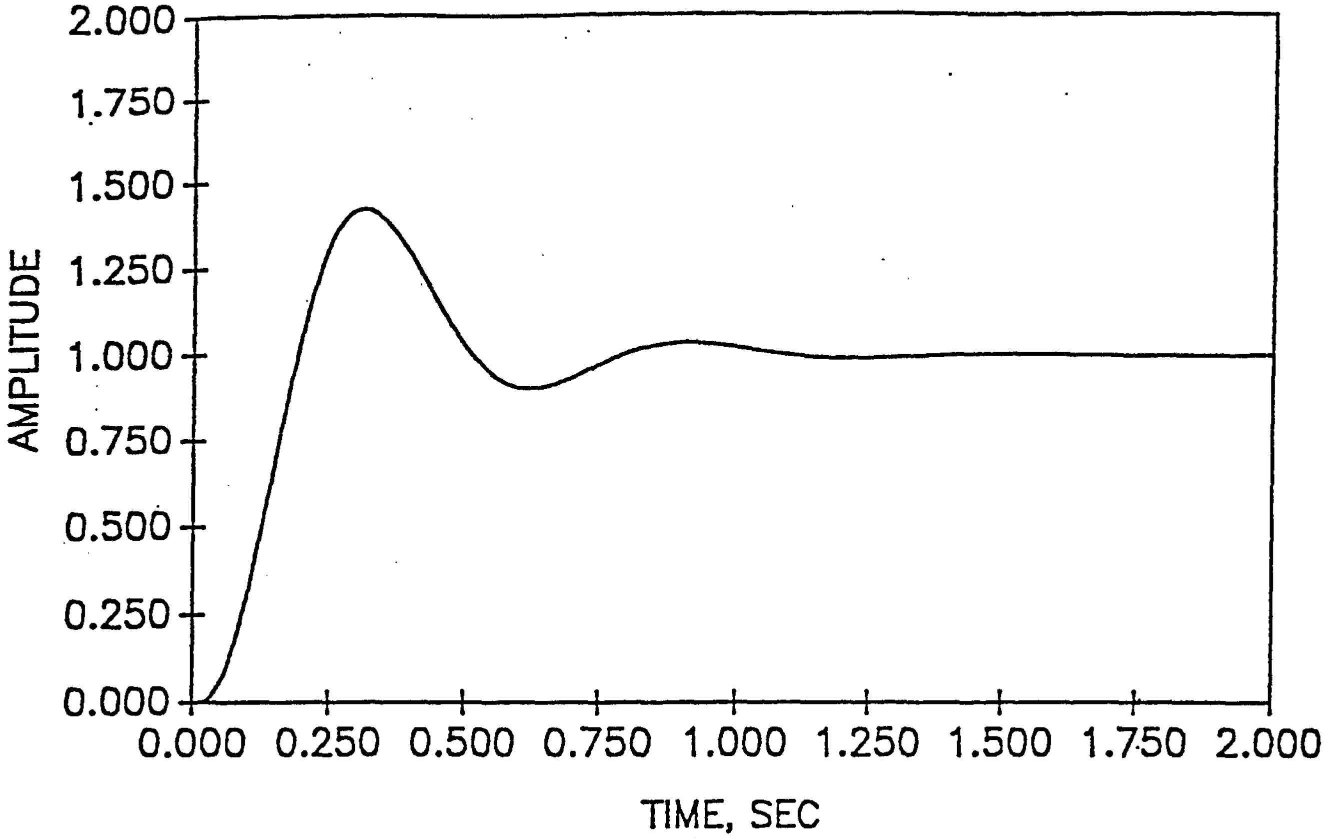


Figure 2-11. Discrete-time closed-loop control system with a gain of 1200.



RAMP RESPONSE OF COMPENSATED SYSTEM

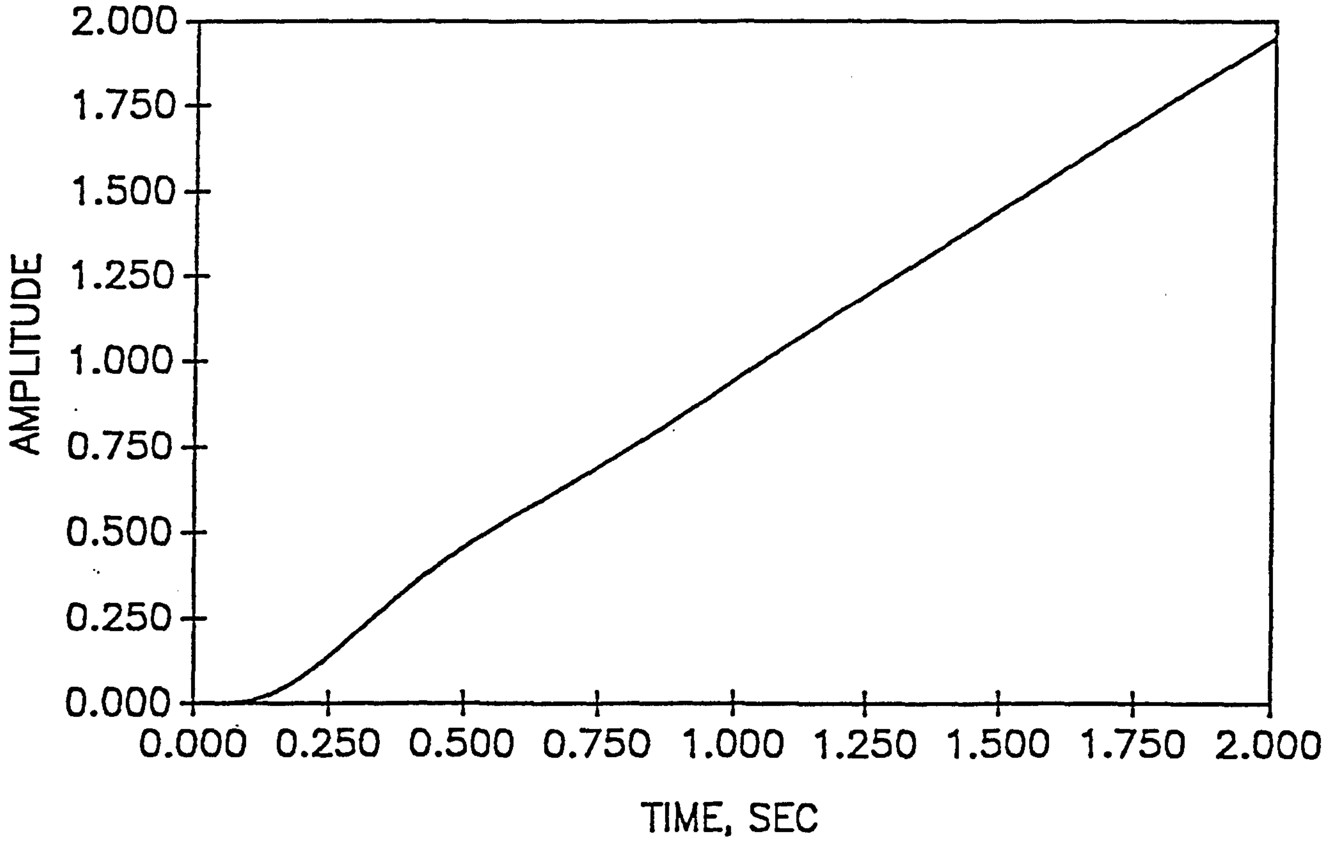


Figure 2-12. Transient response of a discrete-time closed-loop control system for a sample time of 5 milliseconds.

2.11 Summary

A brief overview and introduction to some of the methods of classical control system design have been provided. System representations, the analysis of continuous- and discrete-time single-input, single-output constant-coefficient dynamic systems, the reasons for using feedback, classical control system performance measures, and classical methods for improving the performance of control systems have been discussed.

Two examples which indicated several approaches to the classical design of a closed-loop control system were presented. The first example applied a Bode plot method to develop a lead compensator for a continuous-time system. The second example applied a bilinear transformation which resulted in a compensation algorithm suitable for implementation in a microprocessor-based discrete-time control system. These examples were intentionally kept simple to introduce the basic concepts and allow a comparison to be made regarding the transient response of both closed-loop systems. The performance of the discrete-time control system was illustrated and was very nearly equal to that of the continuous-time system.

REFERENCES

2.1 Ogata, K., Modern Control Engineering, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1970.

2.2 Kuo, B.C., Analysis and Synthesis of Sampled-Data Control Systems, Prentice-Hall, Inc., Englewood Cliffs, New Jersey.

2.3 Dorf, R.C., Modern Control Systems, Fourth Edition, Addison-Wesley Publishing Company, Reading, Massachusetts, 1986.

This page is intentionally blank.

CHAPTER 3
MODERN CONTROL THEORY

3.1 System Modeling

Modern control theory involves the application of mathematical techniques to analyze, synthesize, and optimize devices for the control of a wide variety of systems. Before designing a means to control any dynamic system, a mathematical model of the dynamic system must be derived. This model must include all variables important to the control problem. The mathematical model of a dynamic system represents the operation of the system over time. A key notion in modern control theory is the use of a state variable model for the dynamic system.

The modeling phase is basic to all applications of modern control theory, and involves both the selection of the dynamic system's components and the development of appropriate mathematical models for each component and their interactions. A preliminary measurement and data processing phase is often required to sufficiently characterize each device or component. The complexity of this preliminary phase depends on the nature of the dynamic system and the overall control objective.

The use of state variable methods has been all pervasive during the last two decades. The state variables of a dynamic system are the smallest set of numbers which define the values of all variables of interest relating to a dynamic system or mathematical model at a particular point in time or space. State variable models are commonly applied in modern control theory to represent dynamic systems and their components. The state variable technique is applicable to systems described by linear or nonlinear continuous-time differential equations or discrete-time difference equations. The main reason for the use of the state variable technique is that it permits the use of matrix algebra and vector notation, resulting in highly compact mathematical descriptions of modern control problems.

The most commonly employed state variable model is the linear differential equation:

$$\frac{dx(t)}{dt} = A\,x(t) + B\,u(t),$$

or its nonlinear time-varying counterpart:

$$\frac{dx(t)}{dt} = f(x, u, t).$$

In these models, x is an n by 1 column vector whose time-varying entries x_1, x_2, ..., x_n are the n state variables of the system, and u is an m by 1 column vector whose time-varying entries u_1, u_2, ..., u_m are the m control inputs.

For the linear differential equation model, A is an n by n square matrix of constants which defines the relationships between the state variables and their derivatives and B is an n by m rectangular matrix which defines the way in which the control inputs affect the derivatives. The derivative of each state variable indicates the way in which the state variable evolves or changes over time.

For the nonlinear time-varying model, f(x, u, t) is a column vector of n functions $f_1(x, u, t)$, $f_2(x, u, t)$, ..., $f_n(x, u, t)$ which models the way in which the state variables change with time, the applied control inputs, and the state variables' mutual interaction.

As an example of the use of state variables, consider the motion of a point-mass particle moving in a vertical plane. This simple model is often used to represent the flight of projectiles, missiles, and other weapons. The particle's motion is completely defined if its position and velocity are known as mathematical functions of time. The horizontal and vertical positions and velocities of the particle are a set of state variables for this dynamic system. Four state variables are required in this example, and Figure 3-1 illustrates the resulting control problem.

The particle moves in the plane under the influence of a constant thrust force, F. The thrust direction is assigned to be the control input. This direction is the angle β in radians measured relative to the horizontal axis. The thrust force F can be resolved along the x and y axes into its vertical and horizontal components, F $\sin(\beta)$ and F $\cos(\beta)$.

Newton's law can be applied to develop a set of dynamic equations which mathematically model this system. Since applied force equals mass times acceleration for each component of the particle's motion, the following equations determine the horizontal and vertical accelerations of the particle as functions of the particle's mass, thrust, and applied control input:

$$\frac{d^2x(t)}{dt^2} = \left[\frac{F}{m}\right] \cos(\beta)$$

$$\frac{d^2y(t)}{dt^2} = \left[\frac{F}{m}\right] \sin(\beta) \ .$$

To fully define the particle's motion, it is necessary to know the position and velocity along both axes. One way to achieve this result is to directly solve the above set of second-order differential equations. A second method, and one which leads to the application of modern control theory to this problem, is to convert the mathematical model into a state-variable format involving a set of first-order nonlinear differential equations.

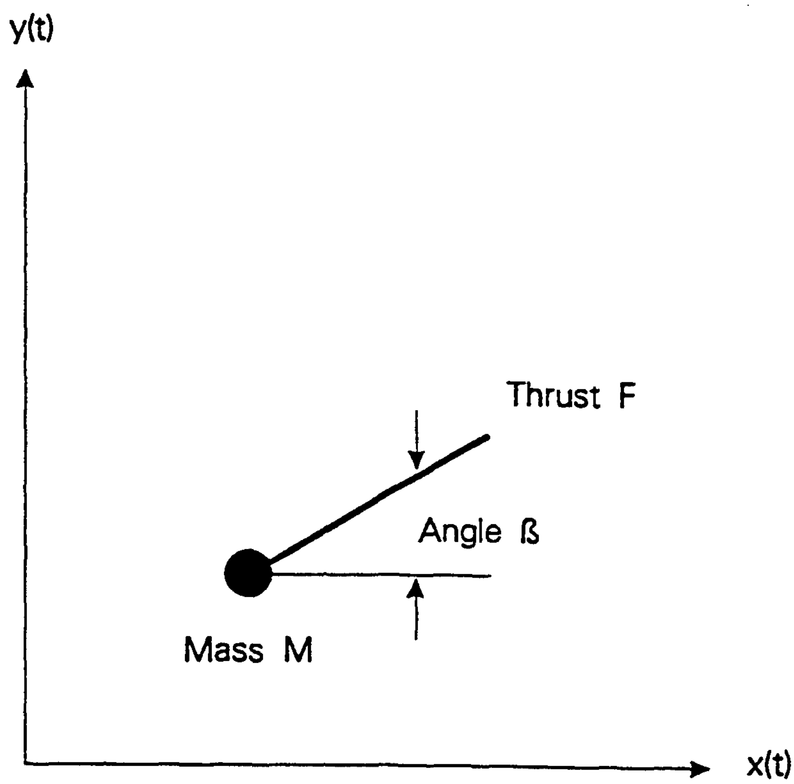


Figure 3-1. Particle motion in the vertical plane.

One method for obtaining the required state variable model is to assign two state variables, x_1 and x_2, to represent the horizontal and vertical positions, and their time-derivatives, x_3 and x_4, to represent the corresponding velocities:

$$\frac{dx_1}{dt} = x_3$$

$$\frac{dx_2}{dt} = x_4 \ .$$

Then, a simple substitution gives:

$$\frac{dx_3}{dt} = \frac{d}{dt}\left[\frac{dx_1}{dt}\right] = \frac{d^2x_1}{dt^2} = \left[\frac{F}{m}\right] \cos(\beta) \quad \text{and}$$

$$\frac{dx_4}{dt} = \frac{d}{dt}\left[\frac{dx_2}{dt}\right] = \frac{d^2x_2}{dt^2} = \left[\frac{F}{m}\right] \sin(\beta) \ .$$

By defining the state variable vector $x = [x_1, x_2, x_3, x_4]$, and the control variable vector $u = [u_1]$ corresponding to the single control input $u_1 = \beta$, and by defining four functions f_1 through f_4 as:

$$f_1(x, u, t) = x_3 \ ,$$

$$f_2(x, u, t) = x_4 \ ,$$

$$f_3(x, u, t) = \left[\frac{F}{m}\right] \cos(u_1) \ , \text{ and}$$

$$f_4(x, u, t) = \left[\frac{F}{m}\right] \sin(u_1) \ ,$$

the motion of the particle can be written in the state variable form:

$$\frac{dx}{dt} = f(x, u, t) \ ,$$

where $f = [f_1, f_2, f_3, f_4]^T$.

The process outlined above has resulted in a notationally compact mathematical model describing the motion of a particle in a vertical plane. This model uses four state variables, and each state variable is defined by a first-order differential equation whose right-hand side involves only the model's parameters, thrust F and mass m, the single control input β, and the four state variables themselves. The variable T is a sample time, T = dt. The notation here indicates that f is based upon a discrete-time state.

If the initial position and velocity of the particle is specified by the state variable vector x(0):

$$x(0) = [x_1(0), x_2(0), x_3(0), x_4(0)]^T ,$$

the motion of the particle can be determined by specifying the control input $u = [u_1]$ as a function of time, and integrating the multi-dimensional state variable differential equation. The state variable differential equation describes the changes in the particle's velocity and position and the way in which the rate of change of velocity depends on the applied control input. For this reason state variable differential equations, or difference equations, are also called state transition equations.

One way in which this model can be implemented in a computer program is by converting the state variable differential equation to a state variable difference equation. A straightforward method involves replacing the derivatives by their approximations:

$$\frac{dx(t)}{dt} \approx \frac{\left(x(t+dt) - x(t)\right)}{dt} \, .$$

Here, x(t) is the value of the state variable vector at time t, x(t + dt) is the value of the state variable vector at a slightly later time t + dt, dt is the small time increment between observations of the state variable vector, and dx(t)/dt is the value of the derivative, or rate of change, of the state variable vector at time t. Rearranging terms, we can obtain the following expression:

$$x(t+dt) \approx x(t) + \left[\frac{dx(t)}{dt}\right] \cdot dt \, .$$

The initial conditions are x(0). If the state of the system is observed at a set of time instants indexed by k, where k = 0, 1, ..., and each instant is separated by a sample time T = dt, we can change notation slightly and write:

$$x\left(k+1)T\right) = x(kT) + \left[\frac{dx(kT)}{dt}\right] \cdot T \, ,$$

$$x(0) = \text{specified} \, .$$

Next, we can drop the explicit dependence on the sample time T and again condense the notation:

$$x(k+1) = x(k) + \left[\frac{dx(k)}{dt}\right] \cdot T \, ,$$

$$x(0) = \text{specified} \, .$$

The continuous-time model has now been converted to a discrete-time model suitable for programming on a digital computer. To apply this method to the previous example, it is necessary to define the four components of the discrete-time state variable vector, $[x_1(k), x_2(k), x_3(k), x_4(k)]^T$, the sample time, T, and the matrix-vector derivatives dx(kT)/dt = f(x, u, kT) = f(x, u, k), where f(x, u, k) has the four components:

$$f_1(x,\ u,\ k) = x_3(k)$$

$$f_2(x,\ u,\ k) = x_4(k)$$

$$f_3(x,\ u,\ k) = \left[\frac{F}{m}\right] \cos\left(u_1(k)\right)$$

$$f_4(x,\ u,\ k) = \left[\frac{F}{m}\right] \sin\left(u_1(k)\right) .$$

The applied control input $u(kT) = u(k) = [u_1(k)]$ is indexed by k in a manner identical to the way in which the state variables have been indexed.

The motion of the particle can now be computed by a relatively simple program which inputs the initial position and velocity and the sequence of applied control inputs, and outputs the values of the state variables at the next sample time. This process is fundamental to all applications and implementations of modern control theory.

To summarize, we began by defining a problem of interest, the motion of a particle of constant mass m in a vertical plane under the influence of a constant thrust force of magnitude F and an applied control input β which determines the direction of the thrust. A mathematical model of this dynamic system was derived by applying elementary physics. The original model obtained was a pair of second-order differential equations. Four state variables representing the particle's position and velocity measured along the horizontal and vertical axes were defined. A state variable model involving four first-order differential equations was then developed. This model is appropriate for use in further mathematical analysis, or simulation of the continuous-time dynamic system. The continuous-time model was then converted to a discrete-time model suitable for implementation in a digital computer program.

3.2 The Development of State Variable Methods

The previous example illustrated the way in which state variable modeling leads to a variety of mathematical expressions for a particular dynamic system, and the resulting compact nature of the state variable representation. The process illustrated is fundamental to the application of modern control theory, as virtually all published results in this technology are presented in terms of a state variable model.

The following sections discuss further applications of the state variable method. The historical basis of the state variable method is outlined and the reader is introduced to stability analysis, optimal control, pole placement, controllability, and observability of dynamic systems. All of these concepts and design techniques are applicable to the design of guidance and control systems for tactical guided weapons.

3.3 Stability Analysis of Dynamic Systems

The emphasis on the state variable approach to modeling, analyzing, and synthesizing modern controllers in feedback form originated in the modeling of dynamic system behavior by mathematical systems of ordinary differential equations. The notion of reducing an n^{th}-order differential equation to a set of n simultaneous first-order differential equations is not a new one, having first been introduced by the mathematician Poincaré[3.1] in 1892. Poincaré also introduced the concept of the state variable as a means of representing the past and present behavior of a dynamic system.

Lord Rayleigh[3.2] investigated the stability of the dynamic system defined by the state transition equation:

$$\frac{dx(t)}{dt} = A\,x(t) \ ,$$

$$x(0) = \text{specified} \ ,$$

in 1894. No control input is applied to this dynamic system. The system's response is due purely to the initial conditions x(0). Lord Rayleigh showed that the motion of this freely responding, uncontrolled system could be resolved into a set of n motions, one motion for each state variable in the model.

If the state variable vector is regarded as a point in n-dimensional space, these motions occur along n independent vectors in this multi-dimensional space. These vectors are called the eigenvectors of the system. The eigenvectors depend only on the contents of the matrix A, the dynamic system model. The magnitude of the motion along each eigenvector depends on the initial state of the dynamic system, specified by the initial condition x(0).

The solution to this state transition equation, the state variable trajectory, is given by the following vector equation:

$$x(t) = c_1 e^{l_1 t} \nu_1 + c_2 e^{l_2 t} \nu_2 + c_3 e^{l_3 t} \nu_3 + \ldots + c_n e^{l_n t} V_n$$

where the vector $x(t) = [x_1(t), x_2(t), \ldots, x_n(t)]$ is the state of the dynamic system, the $c_1, c_2, \ldots, c_n$ are constants which depend on the initial conditions, the $v_1, v_2, \ldots, v_n$ are the eigenvectors, and $l_1, l_2, \ldots, l_n$ are called the eigenvalues. The eigenvalues, which may be complex numbers, roughly correspond to a set of time constants for the dynamic system.

If the magnitude of any one state component is not to grow forever, the real parts of the eigenvalues must be negative, ensuring that the exponential terms eventually decay to zero. This result is one indication of the natural stability of a dynamic system described by this state transition equation.

Further investigations into the stability of dynamic systems were advanced by the publication in 1892 of Lyapunov's[3.3] work. Lyapunov's so-called second method is now the principle means for addressing stability questions occurring in the control of nonlinear dynamic systems and the design of adaptive control systems. Lyapunov's second method addresses the stability of an uncontrolled dynamic system without requiring the solution of the state transition equation.

The basic notion behind Lyapunov's second method is that if the rate of change $dE(x(t))/dt$ of the energy $E(x(t))$ of a dynamic system described by the state variable vector $x(t)$ is negative for every possible state $x(t)$, except for a single equilibrium state x_e, that energy will continually decrease and the system will eventually come to rest at the state x_e where the energy attains a minimum value $E(x_e)$. State variable modeling of dynamic systems played an important role in the development and application of Lyapunov's method.

3.4 Optimal Control

In the presentation to this point, no mention was made of the way in which the applied control input u is determined. A second major application of state variable methods lies in the powerful optimization techniques pioneered by Pontryagin[3.4], developer of the maximum principle of optimal control, and Bellman[3.5], developer of the dynamic programming algorithm. These techniques are essential to the application of the branch of modern control theory called optimal control.

The basic problem of optimal control is to select from a set of admissible controls $u(t)$ one particular control input $u^*(t)$ which minimizes (or maximizes) the performance measure:

$$J(u) = \int_{t_0}^{t_1} L\big(x(t),\ u(t),\ t\big)\ dt\ .$$

The operation of the dynamic system is described by a state transition equation having the form:

$$\frac{dx(t)}{dt} = f\big(x(t),\ u(t),\ t\big)\ ,$$

$$x(t_0) = \text{specified}\ .$$

Here, t_0 is the starting time of the control interval (which may be $t = 0$), t_1 is the ending time of the control interval (which may be $t = \infty$), and the function $L(x(t),u(t),t)$ is selected by the control system designer as a measure of the performance desired from the controlled dynamic system. As an example, L might indicate the final distance of a point mass moving in a vertical plane from a desired terminal position. Optimal control theory provides a means for determining the best control input for steering the particle from an initial position and velocity to a desired terminal position and velocity, while simultaneously minimizing the terminal miss distance.

The maximum principle of Pontryagin is an extension of the Hamiltonian approach to variational problems in analytical mechanics. The Hamiltonian function is defined as:

$$H(x, u, p, t) = L(x, u, t) + p^T f(x, u, t) .$$

The function L is taken from the performance measure J(u), the state transition equation yields f(x, u, t), and the n-dimensional vector p(t) is called the costate vector. Here, the superscript T denotes a transpose operation, necessary for the correct performance of the vector-matrix multiplication operation.

Pontryagin's maximum principle states that if an admissible control action u(t) is to be optimal with respect to maximizing the performance measure J(u), it is necessary that the optimal state trajectory, $x^*(t)$, optimal control action $u^*(t)$, and optimal costate trajectory $p^*(t)$ satisfy the following differential equations:

$$\frac{dx^*(t)}{dt} = \frac{\delta H\left(x^*(t),\ u^*(t),\ p^*(t),\ t\right)}{\delta p}$$

$$\frac{dp^*(t)}{dt} = \frac{-\delta h\left(x^*(t),\ u^*(t),\ p^*(t),\ t\right)}{\delta x} ,$$

and that $H(x^*(t), u^*(t), p^*(t), t)$ attain a maximum due to $u^*(t)$. Note that the right-hand side of these ordinary differential equations involves the mathematical operation of partial differentiation. The first equation is simply a restatement of the system state transition equation. The second equation is an additional set of state transition equations for the costates. In the mathematics of optimal control, the costates play a role similar to that of Lagrange multipliers in conventional static optimization problems.

To solve this set of 2n simultaneous differential equations (n for the state variables and n for the costate variables), one must specify 2n boundary conditions. In many optimal control problems, the initial values of the state variables are known, thus supplying n boundary conditions at the time t_0. The remaining n boundary conditions must come from an analysis of the problem, the desired system performance, and any additional terminal constraints. Also note that the 2n simultaneous differential equations do not directly yield a solution for the optimal control input $u^*(t)$. The 2n equations form a set of necessary conditions which the optimal control input must satisfy. If an arbitrary control input satisfies these equations, it is possible, but not guaranteed, that it is truly the optimal control input.

By invoking Bellman's principle of optimality, which states that portions of the optimal trajectory are themselves optimal, a set of sufficient conditions can be established if the performance measure $J(u^*(t))$ satisfies a certain partial differential equation called the Hamilton-Jacobi-Bellman equation. This equation describes the behavior of the performance measure along an optimal state variable trajectory generated by an optimal control input, and serves as a means to check the optimality of an input control $u^*(t)$ derived from application of the maximum principle. The manner in which the Hamiltonian, the Pontryagin's maximum principle, and Bellman's principle of optimality interact to provide a definition of the desired optimal control input will not be pursued here, but will be further detailed later in this review.

3.5 The Linear Quadratic Regulator

For certain classes of dynamic systems, it is possible to derive explicit formulas for the optimal control input. This process also relies on a state variable model for the system. In 1969 Kalman[3.6] derived a rigorous solution for the linear quadratic regulator control problem. In this problem, the dynamic system to be controlled is described by a linear constant coefficient state transition equation having the form:

$$\frac{dx(t)}{dt} = A\,x(t) + B\,u(t) \, ,$$

$$x(0) = \text{specified} \, .$$

The number of state variables is n and the number of control inputs is m. The matrix A has n rows and n columns, and the matrix B has n rows and m columns.

The performance measure to be minimized is the quadratic functional:

$$J(u) = \int_{t=0}^{t=\infty} \left[x^T Q x + u^T R u\right] dt \, ,$$

where Q is an n by n square matrix of constants and R is an m by m square matrix of constants. These constants are selected by the control system designer to reflect the importance of the various terms in the resulting quadratic expression. In most applications, the goal is to drive the state of the dynamic system as close to the multi-dimensional point $x = 0$ as possible, and hold the value of the state variables at that point. A control problem of this form is called a linear quadratic regulator problem. Autopilots for tactical guided weapons represent a direct application of this design approach. The goal in designing an autopilot is to devise a feedback control system which will maintain the state variables of the airframe as close to a desired reference point as possible over the flight trajectory.

The solution to this problem is a controller based on state variable feedback which takes the form:

$$u(t) = F\,x(t)\ ,$$

$$\text{where } F = -R^{-1}B^{T}K\ .$$

The feedback matrix F has m rows and n columns, and the n by n gain matrix K is the solution of the algebraic Ricatti equation:

$$0 = A^{T}K + KA - KBR^{-1}B^{T}K + Q\ .$$

Considerable effort has been devoted to finding efficient techniques for solving this equation and obtaining numerical values for the elements of the gain matrix K. These values depend on the state transition equations, in terms of the matrices A and B, and on the designer-selected weighting matrices Q and R.

The dynamic system described by the matrices A and B must be controllable, in the sense that there indeed exists a control u(t) which will drive the system to the desired state $x = 0$ in a finite time, and observable, in the sense that all n state variables contribute to the performance measure J. If these conditions are met, then a suitable gain matrix K can be determined as the solution of the algebraic Ricatti equation. The resulting closed-loop feedback system is asymptotically stable, and for any initial condition x(0) the system state will eventually reach the desired state $x = 0$.

The designer must choose the weighting matrices Q and R to reflect the trade-off between penalizing excursions of the state variables from the desired state $x = 0$, and the desire to limit the applied control action by assigning a penalty x^TQx.

The optimal linear regulator outlined above can stabilize an initially unstable system, be designed to realize a prescribed multi-dimensional transient response, and provide a measure of robustness necessary to deal with variations in the system dynamics due to variations in the matrices A and B.

The solution to the linear quadratic regulator problem is important because it provides a methodology for designing closed-loop feedback control systems for an important class of control problems. The method has been extended to time-varying problems, and also provides a basis for many optimal control computational algorithms. The use of the state variable format is essential to both the development and application of this important tool.

This method can also be applied to nonlinear dynamic systems, such as a missile airframe, which are required to be stabilized about a nominal operation point. By linearizing the nonlinear differential equations defining the dynamic system's operation, a linear constant coefficient model can be developed in which the excursions about the reference point serve as the state variables. The linear quadratic solution can then be applied directly to stabilize the dynamic system.

3.6 A Linear Quadratic Regulator Example

An example will help to clarify this process, illustrating the use of the state variable format and the way in which a closed-loop control system can be designed to stabilize a dynamic system. The dynamic system under consideration is shown in block diagram form in Figure 3-2. This simple system consists of a series connection of two integrating units. The input to the first integrator is an externally supplied control signal. This signal is integrated twice and the result appears at the output of the second integrator. The system is inherently unstable since a bounded control input such as a unit step can in some cases produce an unbounded output, a ramp function in this case. The open-loop transfer function has a pair of poles at the origin, s = 0, of the complex plane.

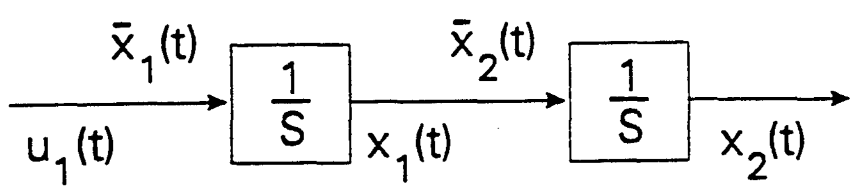


Figure 3-2. Block diagram of an unstable dynamic system.

The two state variables are the outputs of the integrators, identified as x_1 and x_2. The initial performance of this system is considered unsatisfactory because an undamped response results for a step input signal or a disturbance signal. The state transition equation which represents this system is:

$$\frac{d}{dt}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix}[u_1]$$

where the matrix $A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$ and

the matrix $B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$.

The performance measure to be minimized is:

$$J(u) = \int_{t=0}^{t=\infty} \left[x^T Q X + u^T R u\right] dt ,$$

where Q = I is a 2 by 2 identity matrix, and R = r is a one-dimensional scalar matrix. The value of r is selected by the designer to contrast the expenditure of the dynamic system's control energy, measured by the term $u^T Ru$ in the performance measure, with the importance of maintaining the values of the state variables near zero, measured by the term $x^T Qx$.

State variable feedback will be used to design a control system for this dynamic system and obtain a stable response. The state variable feedback is represented by the matrix equation:

$$u = F x ,$$

where $F = -R^{-1} B^T K = -\left(\frac{1}{r}\right) B^T K$,

and the steady-state Ricatti equation is:

$$0 = A^T K + K A - K B R^{-1} B^T K + I .$$

When r = 1 and this equation is solved for the matrix K, the result is:

$$K = \begin{bmatrix} \sqrt{3} & 1 \\ 1 & \sqrt{3} \end{bmatrix} ,$$ and

$$F = \begin{bmatrix} -1 \\ -\sqrt{3} \end{bmatrix} .$$

The resulting stabilizing feedback control system is shown in Figure 3-3. The resulting closed-loop system has a complex pair of poles at:

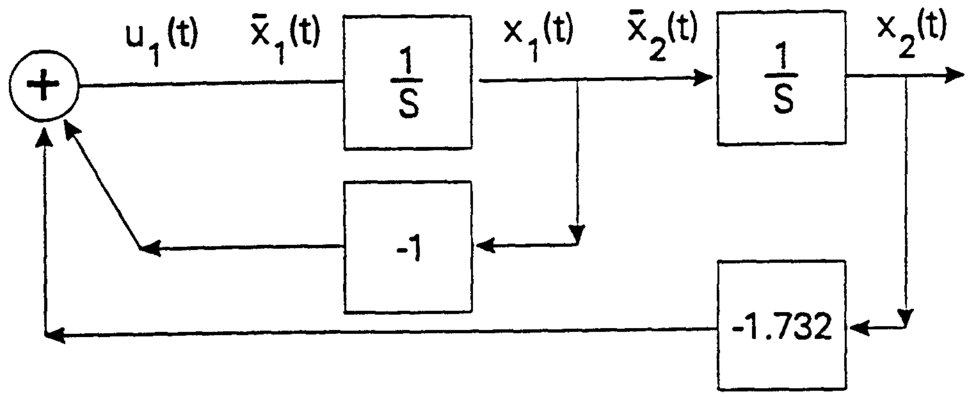


Figure 3-3. Stabilizing feedback control system.

$$\frac{-\sqrt{3} + \sqrt{-j}}{2},$$

an undamped natural frequency of 1 radian per second and a damping factor of 0.86.

The stable transient response of the controlled dynamic system when the initial conditions are $[1,0]^T$ can be determined analytically:

$$x_1(t) = -2.0\,e^{-0.866t}\sin(0.5t-30°) + 2.0\sqrt{3}\,e^{-0.866t}\sin(0.5t)$$

$$x_2(t) = -2.0\,e^{-0.866t}\sin(0.5t)\ .$$

By specifying the performance measure, the designer is thus able to obtain in a direct manner a closed-loop control system which stabilizes the system about the operating point $[0, 0]^T$. The use of the state variable formulation and Kalman's solution to the linear quadratic control problem has eliminated the need for trial and error solutions to this design problem. The control system design which results is, however, dependent on the designer's choices for the weighting matrices R and Q.

3.7 State Variable Feedback and Controllability

When state variable feedback, u = Fx, is selected as a means for regulating the state of a linear dynamic system about a reference point such as the origin, the process of pole placement, or eigenvalue assignment, can be applied. This method provides an alternative design approach, and the resulting closed-loop control system can be different than that obtained by means of the linear quadratic design approach.

The nature of a linear system's transient response is determined by the eigenvalues and eigenvectors associated with the closed-loop system:

$$\frac{dx}{dt} = (A+B\,F)\,x\ .$$

As reported by Kailath[3.7], Popov and Wohnam established a fundamental theorem indicating the ability of a designer to achieve arbitrary eigenvalue assignment by the choice of an m by n feedback gain matrix F. They assumed that complex eigenvalues occur in conjugate pairs, as is always the case in a physical system, and showed that there is a real-valued feedback gain matrix F which allows the eigenvalues of the closed-loop system (A+BF) to take on arbitrarily assigned values if and only if the original linear system defined by the matrices A an B is controllable.

Controllability refers to the capability to transfer the state of a system to the origin, or any other point desired, in a finite time period. The concept of controllability plays an essential role in the application of state variable methods in modern control theory. The solution to a particular control problem may not exist if the dynamic system is not controllable. Most physical systems are fortunately controllable (and observable), but their mathematical models may not possess these desirable properties. For that reason it is necessary to test the model of each system to determine whether or not the model itself is controllable.

Results concerning the controllability of linear time-invariant systems are now readily available and easy to use. The controllability of a linear time-invariant system can be determined by several tests, one of which is to test the rank of the composite n by n • m matrix:

$$\operatorname{rank}\left[B\ :\ AB\ :\ \dots\ :\ A_{n-1}\,B\right] = n\ .$$

A matrix, C, is said to have a rank of n if there exists an n by n submatrix of C, which we will call M, such that the determinant of M is nonzero, and the determinant of every r by r submatrix of C, where $r \geq n + 1$, is zero. As an example, consider the dynamic system specified by the state transition equation:

$$\frac{dx}{dt}\begin{bmatrix}1 & 1\\ 0 & -1\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix} + \begin{bmatrix}1\\ 0\end{bmatrix}\begin{bmatrix}u_1\end{bmatrix}.$$

The structure of this dynamic system is shown in Figure 3-4. The required composite matrix is formed from the matrix B and the matrix product AB:

$$\operatorname{rank}\left[\begin{bmatrix}1\\ 0\end{bmatrix} : \begin{bmatrix}1 & 1\\ 0 & -1\end{bmatrix}\begin{bmatrix}1\\ 0\end{bmatrix}\right] = \operatorname{rank}\begin{bmatrix}1 & 1\\ 0 & 0\end{bmatrix} = 1\ .$$

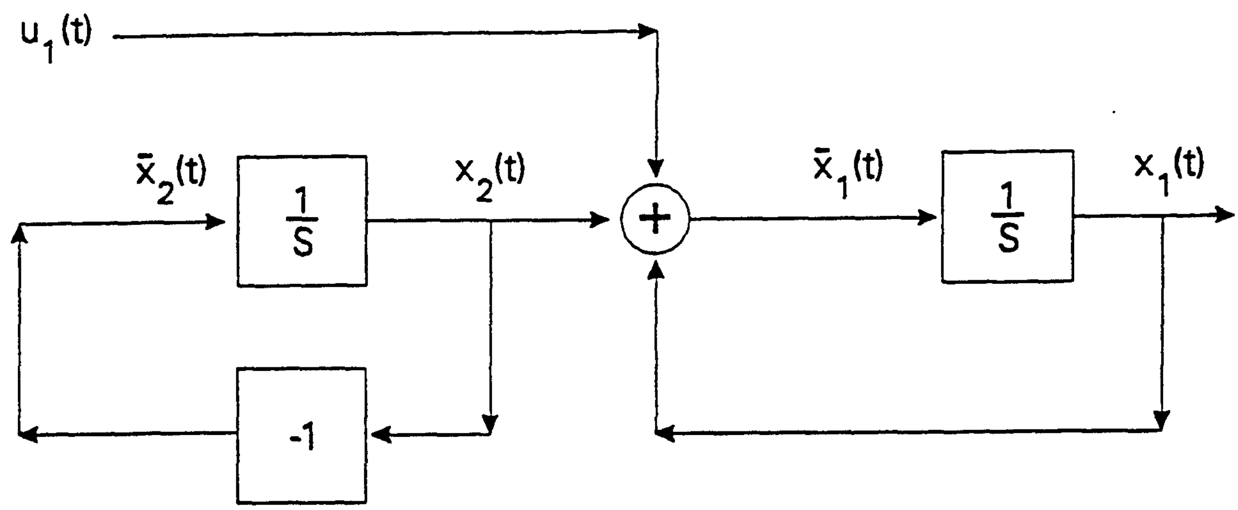


Figure 3-4. Uncontrollable dynamic system.

Since this dynamic system has two state variables, n equals 2. Since the rank of the composite test matrix is 1, this system is uncontrollable. It will not be possible to determine a constant feedback gain matrix which will allow this system to be stabilized about the origin for all arbitrary initial conditions. The physical reason for this result is clear from Figure 3-4. The control input has no effect on the evolution of the state variable $x_2(t)$.

On the other hand, consider a different dynamic system defined by the following linear time-invariant state transition equation:

$$\frac{dx}{dt} = \begin{bmatrix} 1 & 1 \\ 2 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} [u_1] .$$

The structure of this dynamic system is shown in Figure 3-5. The required composite matrix is formed from the matrix B and the matrix product AB:

$$\text{rank} \left[\begin{bmatrix} 0 \\ 1 \end{bmatrix} : \begin{bmatrix} 1 & 1 \\ 2 & -1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} \right] = \text{rank} \begin{bmatrix} 0 & 1 \\ 1 & -1 \end{bmatrix} = 2 .$$

Since there are two state variables, n equals 2. Since the rank of the composite test matrix is 2, this system is controllable and it will be possible to determine a constant feedback gain matrix which will allow this system to be stabilized about the origin for all arbitrary initial conditions.

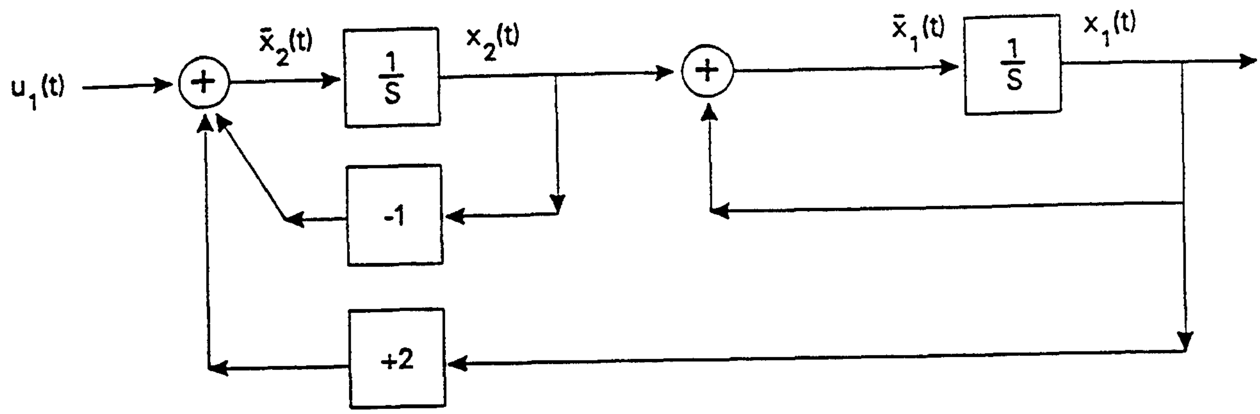


Figure 3-5. Controllable dynamic system.

Controllability theorems for nonlinear dynamic systems which apply in general cases are not yet available. Rather, the designer must select an operating point for the nonlinear system and linearize the dynamic system about that operating point, developing in the process a linear state variable model. The controllability of the linearized model is then used as a surrogate for the controllability of the nonlinear system, under the added assumption that excursions of the system state away from the operating point are kept as small as possible. The requirement of controllability can also be weakened to a requirement of stabilizability, which only requires that the unstable states of a system be controllable.

3.8 Pole Placement or Eigenvalue Assignment

Once it has been determined that the dynamic system in question is controllable, the construction of a feedback gain matrix F which has the property that all of the eigenvalues of the closed-loop system defined by $[A + BF]$ have negative real parts, indicating an asymptotically stable system, can be initiated.

The overall speed of response of the closed-loop linear system is determined by the placement of the system's poles or equivalently, the values of the system's eigenvalues. The shape of its response depends to a great extent on the closed-loop eigenvectors. For a single-loop system, specification of the one closed-loop pole defines a unique system. For a multivariable system, specification of the n closed-loop poles does not define a unique system. The designer has an added capability to choose a set of appropriate eigenvectors and improve the performance of the resulting system. Kailath[3.1] has specified necessary and sufficient conditions for the required gain matrix F to exist, and has outlined a procedure for computing F.

To illustrate the process of placement we will use the unstable open-loop system shown in Figure 3-6. This controllable system is defined by the following state transition equation:

$$\frac{dx}{dt} = \begin{bmatrix} 1 & 1 \\ 2 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} [u_1] \, .$$

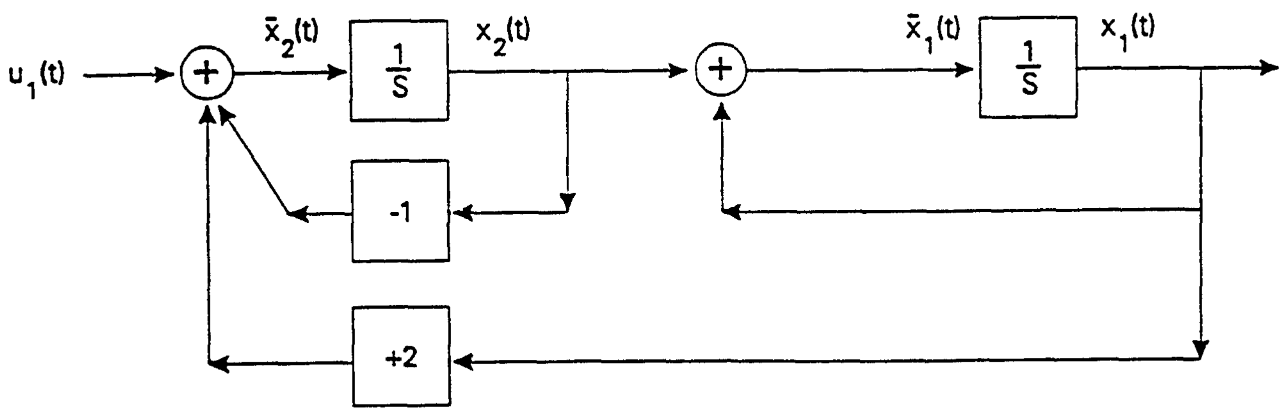


Figure 3-6. Unstable open-loop system ($s_1 = +3$, $s_2 = -3$).

The poles of the open-loop transfer function are at $s_1 = +3$ and $s = -3$ in the complex plane. Instability is indicated by the pole at $s = +3$.

A feedback control law of the form $u_1 = [K_1 \; K_2][x_1 \; x_2]^T$ will be used to relocate the poles of the open-loop system and achieve a stable response for any initial conditions:

$$\frac{dx}{dt} = \begin{bmatrix} 1 & 1 \\ 2 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} [K_1 \; K_2] \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} ,$$

$$\frac{dx}{dt} = \begin{bmatrix} 1 & 1 \\ (K_1+2) & (K_2-1) \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \, .$$

The characteristic equation, the denominator of the resulting open-loop transfer function is determined by:

$$D(s) = \det \begin{bmatrix} (s-1) & -1 \\ (K_1+2) & \left(s-(K_2-1)\right) \end{bmatrix} ,$$

$$D(s) = s^2 + s\left(-K_2\right) + \left(K_2-K_1-3\right) .$$

If, as an example, the designer requires the roots of this characteristic equation to be located at $s = -0.5$ and $s = -1.0$, thus producing a stable system, the required characteristic equation is:

$$D(s) = (s+0.5)(S+1) = s^2 + 1.5s + 0.5 ,$$

and by comparing the coefficients of these two characteristic equations the required feedback gains $K_1 = -1$ and $K_2 = -1.5$ can be algebraically determined. The resulting system stabilized by state variable feedback is shown in Figure 3-7. Note that in this figure we have also indicated the output of the dynamic system as $y_1(t)$.

Methods for pole placement involving higher order systems require the use of a computer-aided design system. Algorithms, procedures, and examples of pole placement continuous and discrete-time systems can be found in Brogan[3.8] and Franklin, Powell, and Ennami-Naeini[3.9]. A method proposed by Bryson and Luenberger[3.10] involves transforming the system:

$$\frac{dx}{dt} = A x + B u$$

into the Luenberger multivariable companion form:

$$\frac{dx'}{dt} = A' x' + B' u$$

by means of an invertible state transformation:

$$x' = Qx$$

where $A' = QAQ^{-1}$

and $B' = QB$

are sparse matrices containing zeros, ones, and other nonzero elements. Then a simple method allows the designer to assign eigenvalues by choosing F' so that the closed-loop system $[A' + B'F']$ has the desired characteristic polynomial, whose roots, or poles, are the eigenvalues of the transformed system. Transforming this result back into the original state coordinate system by means of $F = F'Q$ gives the desired result.

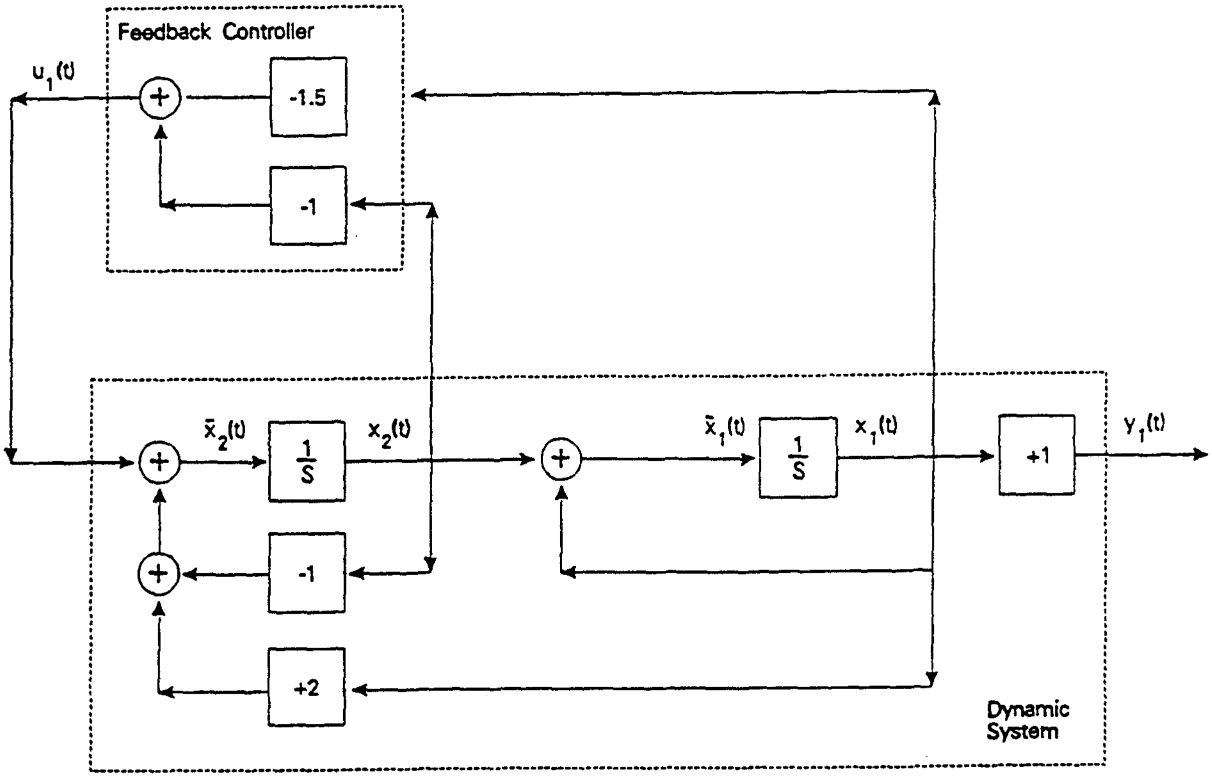


Figure 3-7. System stabilized by state variable feedback ($s_1 = -1.0$, $s_2 = 0.5$).

3.9 The Use of Output Feedback

The linear quadratic regulator method and the methods of pole placement illustrated in the preceding sections have assumed that the complete state variable vector x(t) was available for use, having been measured by suitable transducers or sensors. In many physical systems, only the $m \geq n$ system outputs formed by linear combinations of the system state variables are available. These outputs are formed in the dynamic system represented by the state variable differential and output equations:

$$\frac{dx(t)}{dt} = A\,x(t) + B\,u(t)\ ,$$

$$y(t) = C\,x(t)\ ,$$

where y is an m by 1 vector of output variables whose entries are $y_1, y_2, \ldots, y_m$ and C is an m by n matrix of coefficients. Usually the system is observable in the sense that it is possible to determine the system's initial state x(0), and, by means of the dynamic equations, the system's present state x(t), based on measurements of the current output y(t).

Results concerning the observability of linear time-invariant systems are available and easy to use. The observability of a linear time-invariant system can be determined by several tests, one of which is to test the rank of the composite n by m • n observability matrix:

$$\begin{bmatrix} C \\ CA \\ \cdots \\ CA_{n-1} \end{bmatrix} .$$

If the rank of this matrix equals n, the number of state variables, then the dynamic system is said to be completely observable, and the system state variables can be determined based on measurements of the system output.

A matrix D is said to have a rank of n if there exists an n by n submatrix of D, called M, such that the determinant of M is nonzero, and the determinant of every r by r submatrix of D, where $r \geq n + 1$, is zero.

Consider the linear time-invariant system defined by:

$$\frac{(dx(t)}{dt} = \begin{bmatrix} 1 & 1 \\ 2 & -1 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} [u_1(t)]$$

$$y(t) = [1 \;\; 0] \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} .$$

The test for the observability of this system is:

$$\text{rank} \begin{bmatrix} C \\ CA \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} = 2 .$$

Thus this system is completely observable, and an output feedback controller having $m = 1$ of its poles arbitrarily placed by the designer can be developed. Complex poles can be placed in pairs. Letting $u_1(t) = F\, y_1(t)$, where F is a 1 by 1 scalar matrix yields:

$$\frac{dx(t)}{dt} = \begin{bmatrix} 1 & 1 \\ 2 & -1 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} [F] \, [1 \;\; 0] \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix}$$

$$\frac{dx(t)}{dt} = \begin{bmatrix} 1 & 1 \\ (F+2) & -1 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} .$$

The characteristic equation of this dynamic system is:

$$D(s) = s^2 - (F+3) = 0 ,$$

and the resulting poles are located at $s_{1,2} = \pm \sqrt{(F+3)}$.

If, for example, the designer selects a value for F equal to -2, the poles are located at $s_1 = +1$ and $s_2 = -1$. The resulting system with output feedback is unstable due to the presence of a pole at $s = +1$ in the right hand complex plane. The resulting dynamic system is shown in Figure 3-8.

3.10 State Variable Observers

If the dynamic system has been determined to be controllable and observable, but the complete state is unaccessible, perhaps due to the lack or cost of suitable instrumentation, a feedback control system based on the linear quadratic regulator or pole placement method can still be designed by using an observation $x'(t)$ of the true complete state generated by a state variable observer. The Luenberger state variable observer[3.10] is an auxiliary dynamic system implemented by the control system designer and attached to the original dynamic system.

The Luenberger observer is driven by the available dynamic system state variables, the input to the dynamic system, and the dynamic system outputs. For a dynamic system described by the following state transition equations:

$$\frac{dx(t)}{dt} = A\,x(t) + B\,u(t) ,$$

$$y(t) = C\,x(t) ,$$

a full state variable observer is defined by the state transition equation:

$$dx_c(t) = A_c X_c(t) + B_c y(t) + B\,u(t) ,$$

where $A_c = (A - B_c C)$.

By selecting the matrix B_c the designer determines the eigenvalues of the matrix A_c and thus the asymptotic performance of the observer. A full-order state variable observer generates an observation of the full n-dimensional state vector. A reduced-order observer generates an observation of less than n of state variables. Methods for designing full and reduced-order observers for

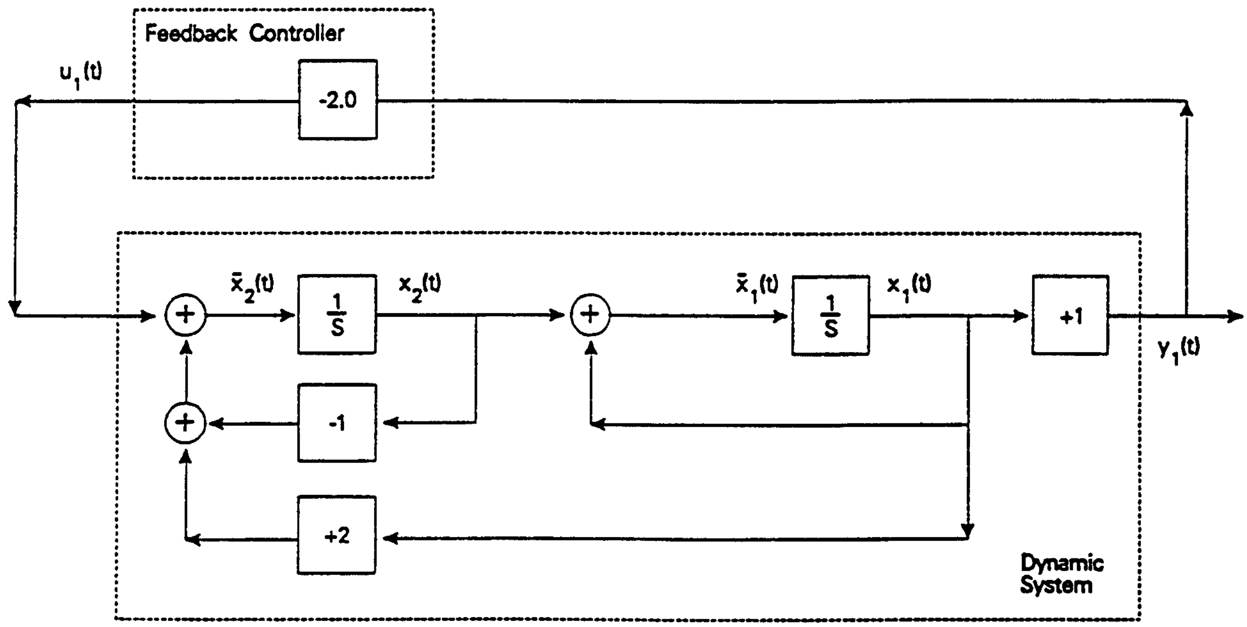


Figure 3-8. Unstable closed-loop system with output feedback ($s_1 = -1.0$, $s_2 = +3.0$).

arbitrary-order linear constant-coefficients are detailed in Brogan[3.8]. As an example of the general design method we will use the dynamic system defined by the following state transition and output equation:

$$\frac{dx(t)}{dt} = \begin{bmatrix} 1 & 1 \\ 2 & -1 \end{bmatrix} \begin{bmatrix} x_1(t) & x_2(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \begin{bmatrix} u_1(t) \end{bmatrix}$$

$$y(t) = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} .$$

Letting $B_c = [\ B_1\ B_2\ ]$ and substituting into the full state variable observer equation we have:

$$A_c = A - B_{cC} = \begin{bmatrix} 1 & 1 \\ 2 & -1 \end{bmatrix} \begin{bmatrix} B_1 & 0 \\ B_2 & 0 \end{bmatrix} = \begin{bmatrix} (1-B_1) & 1 \\ (2-B_2) & -1 \end{bmatrix} ,$$

and the characteristic equation for the observer is then given by:

$$D(s) = \det\left(sI - A_c\right) = s^2 + B_1 s + \left(B_1 + B_2 - 3\right) = 0 .$$

If, for example, the designer selects the observer poles to be located at $s_1 = -3$ and $s_2 = -2$, the required characteristic equation is:

$$D(s) = s^2 + 5s + 6 \ ,$$

and a direct comparison of coefficients yields the result $B_1 = 5$ and $B_2 = 4$. The resulting closed-loop structure is shown in Figure 3-9. Note that the input to the state variable feedback controller is now the output of the observer, rather than a pair of directly measured state variables. The input to the observer consists of the dynamic system output $y_1(t)$ and the signal generated by the feedback controller, $u_1(t)$.

When one or more of the state variables can be directly measured or determined by an algebraic transformation of the system output vector, it is unnecessary to implement a full-order observer and a reduced-order observer will suffice. One possible method for designing a reduced-order observer is to design the full-order observer, and then implement only that subset of observer equations required. A better approach is to design a reduced-order observer which produces only the required state variable observations.

In the dynamic system represented by:

$$\frac{dx(t)}{dt} = \begin{bmatrix} 1 & 1 \\ 2 & -1 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \begin{bmatrix} u_1(t) \end{bmatrix}$$

$$y(t) = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} \ ,$$

the output $y(t)$ equals the state variable $x_1(t)$, and so a reduced-order state variable observer will suffice. To design a reduced-order observer for this system, the designer partitions the state variables into two subsets containing the available state variables, x_1, and the unavailable state variables x_2:

$$\frac{dx_1(t)}{dt} = x_1(t) + x_2(t) \ ,$$

$$\frac{dx_2(t)}{dt} = 2\,x_1(t) - x_2(t) + u_1(t)$$

$$y_1(t) = x_1(t) \ .$$

In the equation for $dx_2(t)/dt$ the terms involving $x_1(t)$ and $u_1(t)$ are then temporarily treated as known time functions. Also,

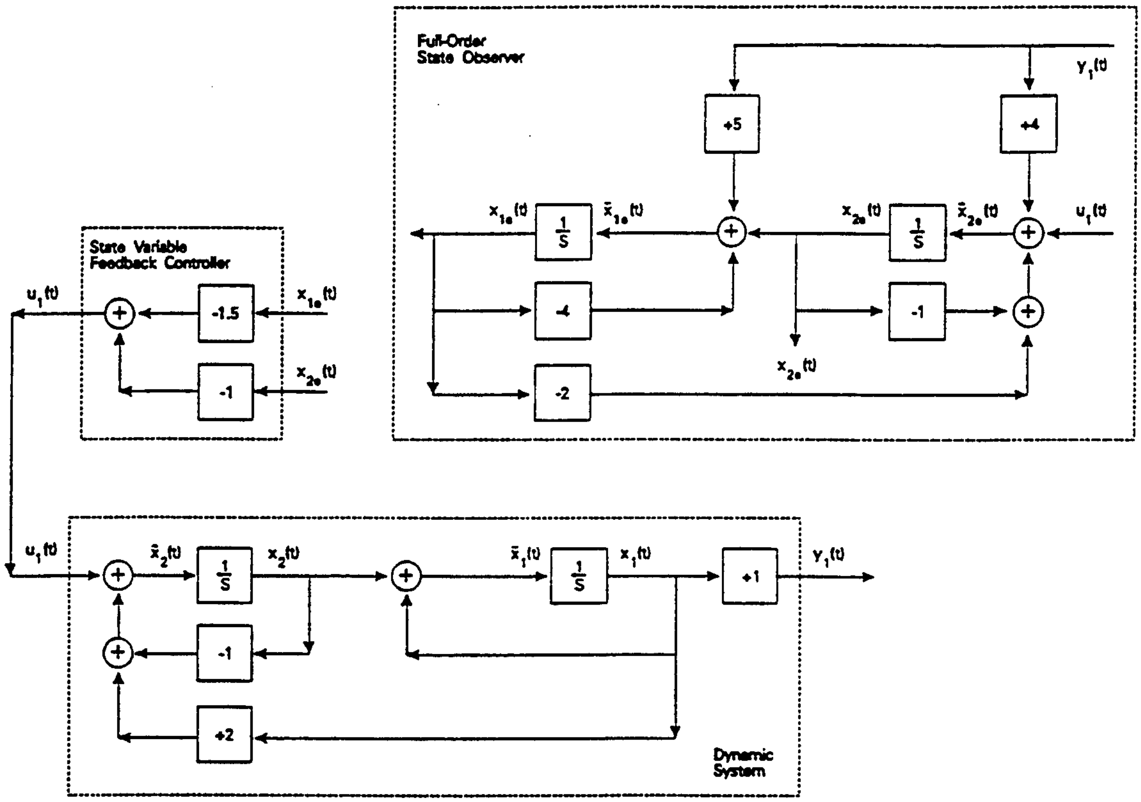


Figure 3-9. System stabilized by state variable feedback
and use of full-order observer.

$$\frac{dy_1(t)}{dt} = \frac{dx_1(t)}{dt} = x_1(t) + x_2(t), \quad \text{or}$$

$$x_2(t) = \frac{dy_1(t)}{dt} - y_1(t) \ .$$

The observer is assumed to have the same dynamic structure as the original dynamic system, and a feedback term based on the state variable error between the observer system and the original dynamic system is added:

$$\frac{dx_{2e}(t)}{dt} = -x_{2e}(t) + 2x_1 + K\left(x_2(t) - x_{2e}(t)\right)$$

$$\frac{dx_{2e}(t)}{dt} = -x_{2e}(t) + 2x_1 + K\left[\frac{dy_1(t)}{dt} - y_1(t) - x_{2e}(t)\right] \ .$$

An observer error is then defined and the appropriate algebraic substitutions made:

$$\frac{de(t)}{dt} = \frac{dx_2(t)}{dt} - \frac{dx_{2e}(t)}{dt} ,$$

$$\frac{de(t)}{dt} = \left(2\,x_1(t) - x_2(t) + u_1(t)\right) -$$

$$\left[-x_{2e}(t) + 2x_1 + K\left[\frac{dy_1(t)}{dt} - y_1(t) - x_{2e}(t)\right]\right] ,$$

$$\frac{de(t)}{dt} = -(1+K)\left(x_2(t) - x_{2e}(t)\right) = -(1+K)\,e(t) .$$

An appropriate pole location can then be selected by choosing a numerical value for the feedback gain K. Generally the observer poles are placed slightly to the left of the dynamic system poles in the complex plane. The resulting reduced-order state variable observer is then defined by the state transition equation:

$$\frac{dx_{2e}(t)}{dt} = -x_{2e}(t) + 2x_1 + K\left[\frac{dy_1(t)}{dt} - y_1(t) - x_{2e}(t)\right] .$$

One difficulty encountered here is the need to develop the derivative of the system output, $dy_1(t)/dt$. This can be overcome by defining an auxiliary state variable:

$$x_3(t) = x_{2e}(t) - K\,y_1(t) \text{ or } x_{2e}(t) = x_3(t) + K\,y_1(t) .$$

Then $\dfrac{dx_3(t)}{dt} = \dfrac{dx_{2e}(t)}{dt} - \dfrac{dy_1(t)}{dt}$, or

$$\frac{dx_3(t)}{dt} = -x_{2e}(t) + 2x_1 - K\left(y_1(t) + x_{2e}(t)\right) ,$$

$$\frac{dx_3(t)}{dt} = -(1+K)\,x_{2e}(t) + (2-K)\,y_1(t) .$$

Now $x_3(t)$ can be computed without the need for a derivative of $y_1(t)$, and the required observation $x_{2e}(t)$ can be computed in terms of $x_3(t)$ and $y_1(t)$. If, for example, the designer selects a feedback gain of $K = 19$, the observer pole is placed at $s = -20$, and the resulting closed-loop control system including the feedback controller is illustrated in Figure 3-10.

If the linear dynamic system is subject to random disturbances or if the measurements of the output vector y(t) are accompanied by random noise, the dynamic state observer is stochastic in nature, and the estimate $x_e(t)$ can be performed in a least-squared-error manner. If all the

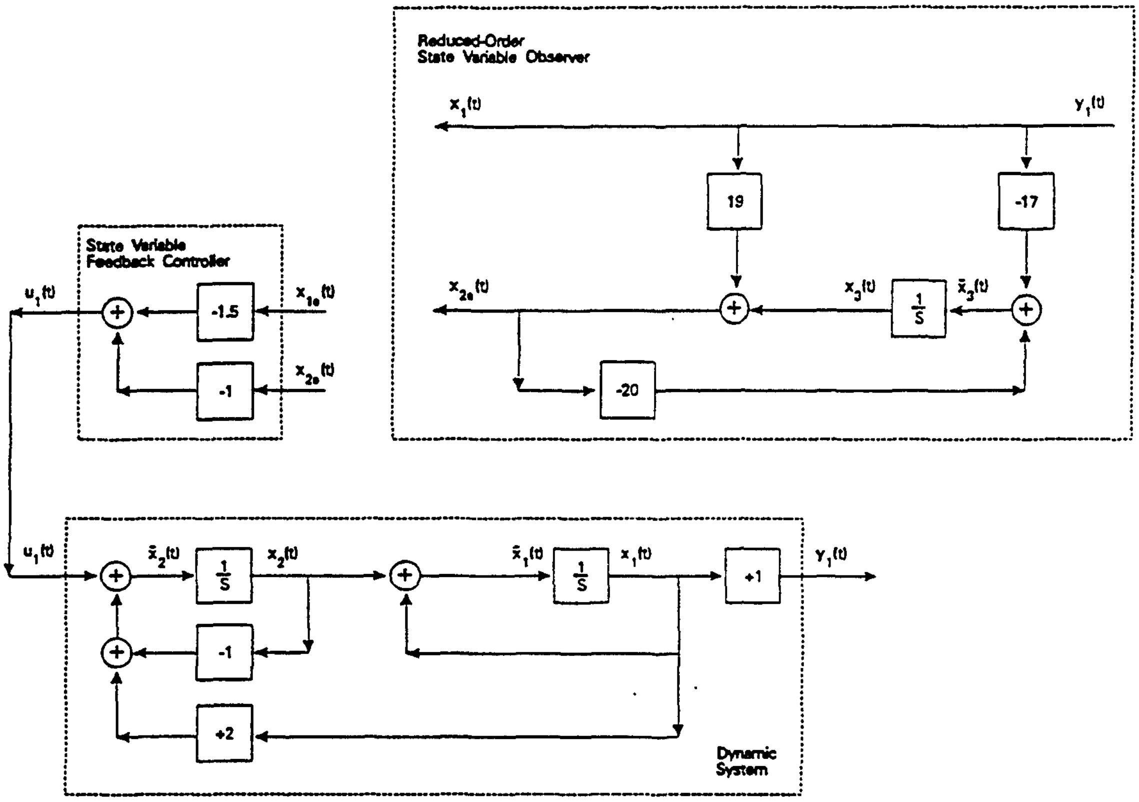


Figure 3-10. System stabilized by state variable feedback
and use of reduced-order observer.

measurements of the output y(t) are corrupted by additive white noise, and the estimator is also of order n and has an observer state variable vector $z(t) = x_e(t)$, the estimator is known as a Kalman filter. More will be said about Kalman filters in a later chapter of this report.

3.11 Summary

This chapter discussed the modeling phase of system development, introduced the state variable modeling method for dynamic systems, and indicated a few useful applications of this method. The examples have included the analysis of system stability, the design of feedback controllers based on the linear quadratic and pole placement methods, and the need for and use of state variable observers which reconstruct the system state variables from measurements of the input and output of the dynamic system. These methods and techniques have direct application to the analysis, modeling, and design of guidance and control systems, particularly autopilots, for tactical guided weapons. Several additional examples indicating the use of these methods will be presented later in this report.

REFERENCES

3.1 Poincaré, via: MacFarlane, A.G.J., The Development of Frequency-Response Methods in Automatic Control, IEEE Transactions on Automatic Control, Vol. TAC-24, pp. 250-265, 1979.

3.2 Rayleigh, Lorel, The Theory of Sound, Republished by Dover, New York, 1945.

3.3 Lyaponov, via: Willems, J.L., Stability Theory of Dynamical Systems, Nelson, London, 1970.

3.4 Pontryagin, L.S., Boltyanskii, V.G., Gamkrelidge, R.V., and Mischenko, E.F., Mathematical Theory of Optimal Processes (in Russian); English translation published by Interscience - Wiley, New York, 1962.

3.5 Bellman, R., Dynamic Programming, Princeton University Press, Princeton, N.J., 1957.

3.6 Kalman, R.E., Falb, P., and Arbib, M.A., Topics in Mathematical System Theory, McGraw-Hill Book Company, New York, 1969.

3.7 Kailath, T., Linear Systems, Prentice-Hall, Englewood Cliffs, New Jersey, 1980.

3.8 Brogan, W.L., Modern Control Theory, Prentice-Hall, Englewood Cliffs, New Jersey, 1985.

3.9 Franklin, G.F., Powell, J.D., and Emami-Naeini, A., Feedback Control of Dynamic Systems, Addison-Wesley Publishing Company, Reading, Massachusetts, 1986.

3.10 Luenberger, via: O'Reilly, J., Observers for Linear Systems, Academic Press, London, 1983.

3.11 More on Pontryagin and Bellman is presented via: Athans, M. and Falb, P.L., Optimal Control, McGraw-Hill Book Company, New York, 1966.

CHAPTER 4
DYNAMIC SYSTEMS

4.1 System Concepts

A system[4.1] is any device, procedure, or scheme which behaves according to a well-defined description. Control systems are described in terms of transfer functions, differential equations, difference equations, and other mathematical constructs. The function of a system is to operate on an input of information, energy, or matter, generally over a period of time, and to yield transformed information, energy, or matter. A general system is illustrated in Figure 4-1. The description of this system's behavior may be a deterministic mathematical model or it may be a stochastic model which involves random parameters or variables.

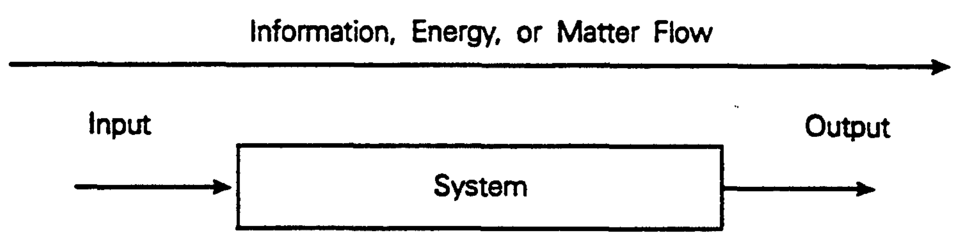


Figure 4-1. Generalized system schematic.

The concept of a dynamic system[4.2] is central to the application of modern control theory. A dynamic system is described as any system whose behavior and description includes or involves mathematical operations which depend on time. These operations may be time delays or lags, differentiation, integration, or the action of time-varying functions. A dynamic system is thus any system whose behavior evolves with or changes over time.

There are two further concepts associated with the notion of a dynamic system, the state of the dynamic system, and the idea of a state transition. The state of a dynamic system is that set of information which allows one to predict the dynamic system's observable behavior. Knowledge of the present state and the manner in which the state evolves, or is transformed, is sufficient to allow accurate predictions to be made about the future state of the dynamic system. The state of a dynamic system is transformed as a result of the passage of time and any external influences. This evolutionary process is called the state transition process, and the mathematical model which describes the state transition process is called the state transition equation.

Dynamic systems have many applications in the analysis and design of tactical guided weapon systems. Usually dynamic systems of interest are described in terms of sets of simultaneous differential or difference equations. For example, the aerodynamic state of a missile involves the missile's position and velocity measured in terms of three linear coordinates (x, y, and z) and three angular coordinates (pitch, yaw, and roll) for a total of six states. Additional states are required to define the operation of the missile's internal systems, the seeker, guidance computer, autopilot, and actuators.

To completely define a dynamic system one must specify the time interval of the system's operation and the way in which time will be measured, the inputs and the outputs of the system, the states of the dynamic system, and the mathematical relationships describing the state transition mechanism.

The passage of time may be measured in a continuous manner, in which case the input, state, and output are all functions of time, indicated by the continuous variable t, or in a discrete manner, in which case the input, state, and output are all functions of the discrete index k. The initial time of interest is usually taken as zero. The final time may be some finite time t_{max} or infinity. In either case the system is referred to as a continuous-time system or a discrete-time system.

In a continuous-time system all quantities are measured continuously over time. In a discrete-time system, all quantities are measured only at discrete points in time. Continuous-time dynamic systems arise naturally in problems of physical mechanics and analog circuitry. Discrete-time systems arise naturally in the development of computer models or simulations of dynamic systems and whenever a digital computer is used to control a process.

The input, state, and output may each consist of a single quantity or scalar, or a vector of multiple quantities. A dynamic system having more than one state is called a multivariable system. The state of any dynamic system represents a history of the applied input and contains all the information necessary to compute the next state and current output of the system based on the current input.

There are several important classes of dynamic systems which have wide application in engineered systems such as tactical guided weapons. If the mathematical equations which define the relationship between the next state, the present state, and the present input are linear, the system is called a linear dynamic system. These equations usually take the form of differential equations or difference equations. Linear dynamic systems are algebraic in nature, and there exists a well-developed theory and body of computational methods for the solution of linear dynamic system problems. An important class of linear systems is the class of time-invariant linear systems in which the state transition equations are linear constant-coefficient differential or difference equations. For these highly-important systems, the Laplace-transform or Z-transform methods may be applied to efficiently solve the state transition equations, finding the system output for any specified system input and set of initial state conditions.

4.2 Dynamic System Problems

There are five general classes of problems which arise in the study and application of dynamic systems:

- implementations
- networks
- simulations
- simplifications
- analysis

Implementation problems involve the realization of a dynamic system which corresponds to some set of system specifications. In general this involves finding a dynamic system which corresponds to a given input-output process. For example, it may be necessary to design a system which automatically tracks a specific input signal with a minimal amount of error. Autopilots and guidance computers are designed based on a set of specifications for the desired performance of a tactical guided missile.

A problem related to implementation is the identification problem, which involves identifying the structure of an unknown dynamic system, or the value of the parameters of a system whose structure is known, based on a comparison of the system's inputs and outputs. The dynamic model of the missile or other airframe used to design an autopilot or guidance system must be identified based on wind tunnel tests or comparisons with other airframes whose characteristics are known.

Network problems involve the construction or composition of a network of dynamic systems to accomplish some specific task, or with the decomposition or restructuring of a given dynamic system into a network of smaller systems. The development of the complex mathematical model for the six-degree-of-freedom motion of a missile airframe and the design of a control system to stabilize and control that motion is an example of a network design problem. The resulting control system consists of a network of interconnected and interacting control components and sensors.

Simulation problems involve the development of alternate models of a given system. As an example, the development of a simulation of a dynamic system may involve the development of a digital computer algorithm which solves the input, state transition, and output relations for a physical system and produces a numerical estimate of the physical system's performance. The development of mathematical models and simulations is an important application of modern control theory to tactical guided weapon design.

The goal in a simplification problem is to find a less complicated model for a specified dynamic system which produces the same results as the original model, or produces results which are in some way good enough to permit use of the simpler model for design or analysis. One simplification technique often used is the approximation of a high-order continuous-time linear dynamic system by a second-order dynamic system whose complex-conjugate poles are identical to the poles of the higher-order system closest to the origin of the complex plane. This allows rapid estimation of the transient response of the more complex system, since results for second-order systems are readily available.

In aerodynamic control system design it is common practice to begin the design of a closed-loop control system by focusing on the airframe motion in the pitch plane and ignoring any airframe motions along the yaw or roll axis. This considerably reduces the problem's complexity by eliminating temporarily a number of state variables.

Analysis problems deal with many other aspects of dynamic systems, including their stability, controllability, and observability. A dynamic system is considered to be completely state-controllable if there is some input function which, if applied at some time t, drives the dynamic system state to the origin at some later time t. A dynamic system is considered to be completely state-observable if the input and output data measured over some time span from t to t′ allow one to uniquely determine the initial state of the system at time t.

4.3 Modeling Dynamic Systems

To apply modern control theory to the guidance and control of tactical weapons, it is necessary to apply the tools of mathematical system analysis and model building. The use of a mathematical model is necessary if one is to investigate and understand the dynamic system's behavior. The mathematical model defines the nature of the dynamic system (linear, nonlinear, time-varying, time-invariant, etc.) and allows the system to be treated and manipulated by mathematical means. Models are required to construct simulations, to develop control algorithms and to investigate and compare overall system performance.

To develop a mathematical model of a dynamic system, the physical variables present in the system must be related by mathematical structures such as differential or difference equations. Concepts for model building are drawn from all areas of science and technology which impact the performance of a tactical weapon. For example, the following element equations define the small-signal performance of the three basic electronic circuit elements:

Element	Defining Equation
Resistor	$v_r(t) = R \cdot i_r(t)$
Capacitor	$i_c(t) = C \cdot dv_c(t)/dt$
Inductor	$v_l(t) = L \cdot di_l(t)/dt.$

In these defining element equations v_r, v_c, and v_l denote the instantaneous voltage across any resistor, capacitor, and inductor and i_r, i_c, and i_l denote the instantaneous current through these circuit elements. Any electronic circuit, regardless of its complexity, containing these three basic elements can be reduced to a mathematical model, a set of simultaneous differential equations, by applying these element equations and the basic laws of circuit theory. Once the model has been developed, the response of the circuit to any input signal can be determined by analysis or simulation.

The mathematical model of an electronic circuit described thus far is a linear, constant-coefficient, time-invariant dynamic system model. The steady-state performance of the dynamic system represented by this model, at such time in the future when all derivatives are zero, indicating that no further changes are occurring in the state of the system represented by the set of element voltages and currents, is a static model represented by a set of linear algebraic equations.

A judicious choice of simplifying assumptions is always required when developing a mathematical model of any dynamic system. When the model is to be treated analytically, these assumptions are required to limit the complexity of the model. This implies a tradeoff, or compromise, between model complexity and accuracy.

Mathematical models for dynamic systems can be classified in a number of ways. A first distinction is between lumped-parameter and distributed-parameter models. In a lumped-parameter model, the physical parameters of the model such as mass, resistance, or capacitance are assumed to be spatially concentrated. This assumption leads to mathematical models consisting of sets of coupled differential or difference equations. In distributed-parameter models, the spatial nature of the problem is explicitly taken into account and this process leads to mathematical systems of partial differential equations of parabolic, elliptic, or hyperbolic type. When dealing with dynamic systems originally described by partial differential equations, two approximations which are frequently made are a discretization in space, resulting in a lumped-parameter model described by a set of coupled differential equations, or discretization in time, leading to a set of coupled difference equations. In the first case, the mathematical model remains a continuous-time model. In the second case, the model is called a discrete-time model.

Models for dynamic systems can also be classified as deterministic or stochastic models. In a stochastic model the relationships between the model's parameters incorporate probabilistic effects due to chance events or randomly occurring changes in the model's structure. A deterministic model does not account for such effects. Deterministic models can also be classified as parametric and non-parametric models. Parametric models include such mathematical constructs as algebraic equations, systems of differential or difference equations, and transfer functions. Parametric models result from a theoretical analysis of the dynamic system's underlying behavior. Non-parametric models result from an experimental analysis of a physical system, and typically consist of tabulated results and observations which serve as a description of the system's behavior.

To develop a mathematical model of a dynamic system by means of a theoretical analysis, the model developer must rely on an ability to decompose the problem into a set of manageable subproblems. To each subproblem the developer applies basic laws of science such as the conservation of energy, mass, and momentum. These basic laws are selected from an array of such laws depending on the technology applicable. By applying these basic laws, a set of coupled equations which provide a reasonable model of the underlying dynamic system is obtained. By carefully selecting the variables describing the system's performance a set of state transition equations can be developed. In most cases of interest these will be in the form of a set of coupled first-order differential or difference equations. These equations, together with any specific initial conditions, will allow the analyst to solve for the performance of the dynamic system in response to any applied set of inputs.

4.4 Summary

This chapter has introduced several important concepts associated with dynamic systems. Dynamic systems, systems whose behavior evolves over time, are fundamental to the application of modern control theory. Several classes of dynamic systems, including deterministic and stochastic systems, have been described. The five major problem types associated with dynamic systems have been highlighted, and mention was made of the way in which each problem type applies to the design and development of tactical guided weapons.

REFERENCES

4.1 Ellis, D.O. and Ludwig, F.J., Systems Philosophy, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1962.

4.2 Ogata, K., Modern Control Engineering, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1970.

This page is intentionally blank.

CHAPTER 5
SYSTEM IDENTIFICATION

5.1 The Basic Identification Problem

The problem of system identification involves building a mathematical model of a dynamic system based on input and output measurements. The general idea is to observe the behavior of the dynamic system over a time interval and, by recording observations of the system's input and output, develop a description of the dynamic system's behavior in the form of a mathematical model.

The mathematical model which results from the process of system identification is then used for other purposes such as predicting the future output of the system or investigating means for controlling the system's operation. The application of both classical and modern control theory assumes that a mathematical model of the underlying dynamic system is available. The applicability of theoretical results thus depends on the development and availability of a satisfactory mathematical model.

In practice there are two main approaches toward the development of a mathematical model for a dynamic system. The first approach, the process of system analysis, divides the dynamic system into conceptually smaller subsystems whose properties are well understood from previous experience, physical laws, and well-established relationships. The mathematical models for these subsystems are then assembled to form a composite model of the overall dynamic system. This analytical approach to mathematical modeling does not involve any direct experimentation on the dynamic system. The system analysis approach is the only one possible when a mathematical model is required for a new or physically nonexistent dynamic system such as a proposed tactical weapon.

The second approach to the development of a mathematical model for an existing dynamic system is to conduct experiments in which the dynamic system being investigated is treated as a black box having unknown contents. The structure of the dynamic system is initially assumed to be unknown. Input signals are applied to the dynamic system and these signals, and the output response they induce over time, are recorded. Analysis of these recorded data allows the experimenter to infer the structure of a mathematical model for the dynamic system. This experimental process of developing a mathematical model for a dynamic system is called system identification. During the process of system identification the complementary approaches of modeling and experimentation are used simultaneously to maximize the information gleaned from identification experiments and to verify the results of experimental data analysis.

The general steps in the system identification procedure are:

- apply a specific set of test inputs to the unknown dynamic system
- collect the corresponding input and output data
- select a set of candidate mathematical models
- pick one member of the candidate model set as the best mathematical model to represent the unknown dynamic system

The operational nature of the experimental system identification procedure is illustrated in Figure 5-1.

In each of these steps, the investigator must be guided by intuition, experience, and the available test data. The data are normally recorded during a specially-designed identification experiment by sampling in discrete time using a digital computer. Some system identification methods require a deterministic test input to be supplied, while others utilize random or pseudo-random input sequences. The overall objective is to extract the maximum information about the structure of the unknown system from the data that have been recorded. The choice of inputs, sampling rates, noise filters, and signals to be measured are all important. For example, the sampling rate must be at least twice as high as the maximum frequency which the system is likely to encounter in practice.

The set of candidate models is selected by the experimenter based on experience in dealing with dynamic systems similar to the unknown system. The system identification process requires the experimenter to select the best member of the candidate model set. Engineering insight, intuition, and a prior knowledge must be combined with a formal modeling approach if good results are to be obtained. If the system identification process is to be implemented manually, the model selected may be graphical, formed by a set of curves relating the system input and output. For automated system identification an analytical model, formed by an assumed mathematical relationship between the test input and test output, is preferred.

Semi-graphical time-domain models are used in classical methods of system identification. These identification methods, which have been developed primarily for classical single-input, single-output, time-invariant dynamic systems, are implemented by applying a unit step or an approximate impulse function (formed by a high amplitude, short duration, rectangular pulse) and recording the system output. Data concerning the impulse or step response of the unknown system are then experimentally obtained.

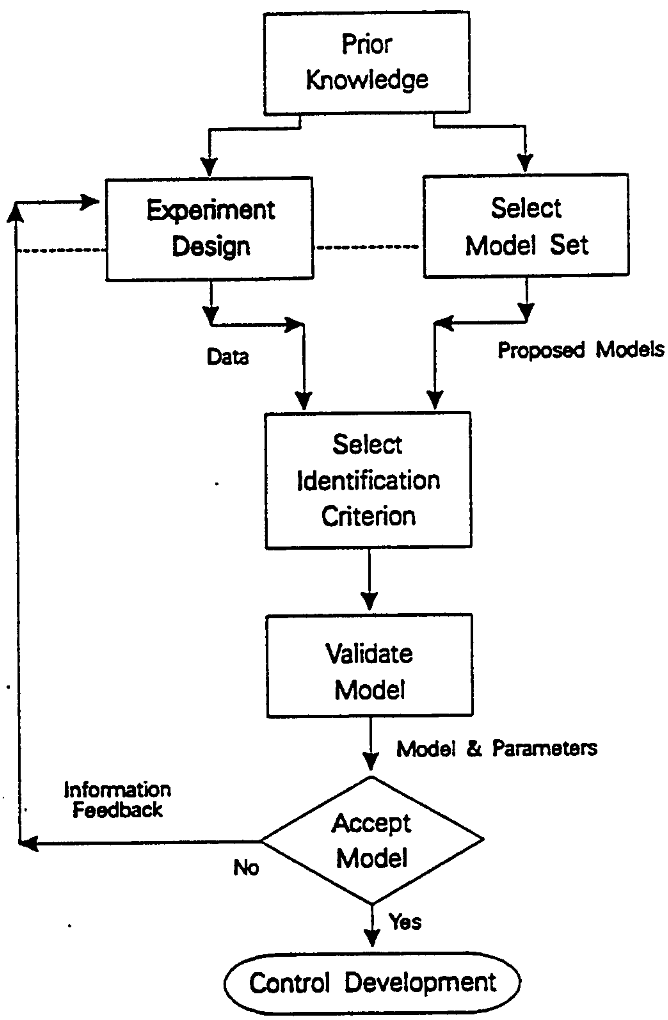


Figure 5-1. System identification procedure.

Information about the gain of the system transfer function and the dominant pole locations is then derived by analysis of the recorded data. The observed step or impulse response can often be approximated by the response of a low-order model, and the Laplace transform of the impulse response for this simpler model yields the unknown system transfer function. The impulse response of a linear time-invariant system can also be obtained via the cross-correlation between the output and the input when the input is white noise.

Figure 5-2 shows the transient response of a second-order linear time-invariant system subjected to a unit step input. The parameter which distinguishes each curve is the damping factor ξ, and the horizontal axis is scaled to ω_n*t, where ω_n is the natural frequency of the system and t is the elapsed time in seconds. This dynamic system is represented by the transfer function:

$$G(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} .$$

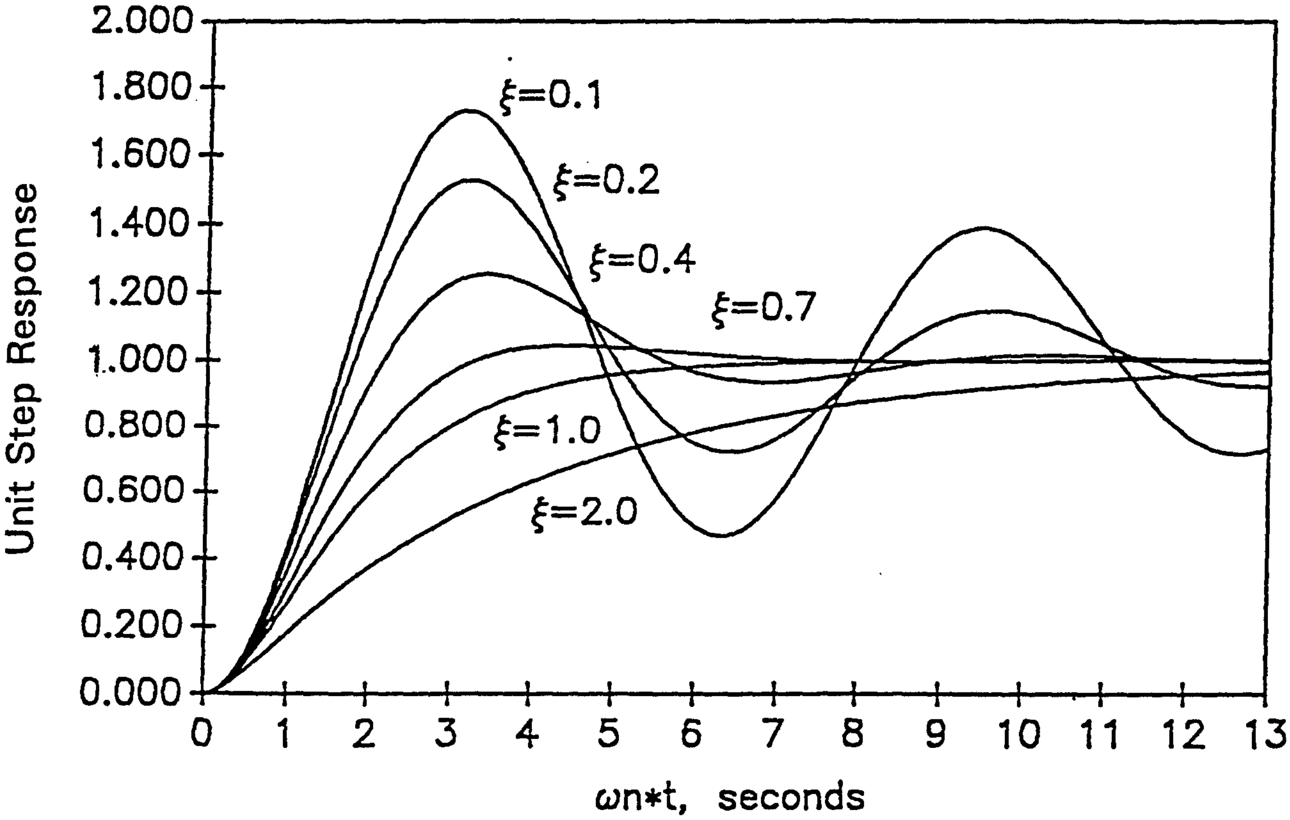


Figure 5-2. Transient response of a second-order system.

The unit step response of a general control system is illustrated in Figure 5-3. Several time domain specifications are labeled in this figure, including the peak time, the maximum overshoot, the steady-state error, the rise time, and the settling time. These quantities can be determined for an arbitrary single-input, single-output stable control system, or dynamic system by analyzing the system's step response.

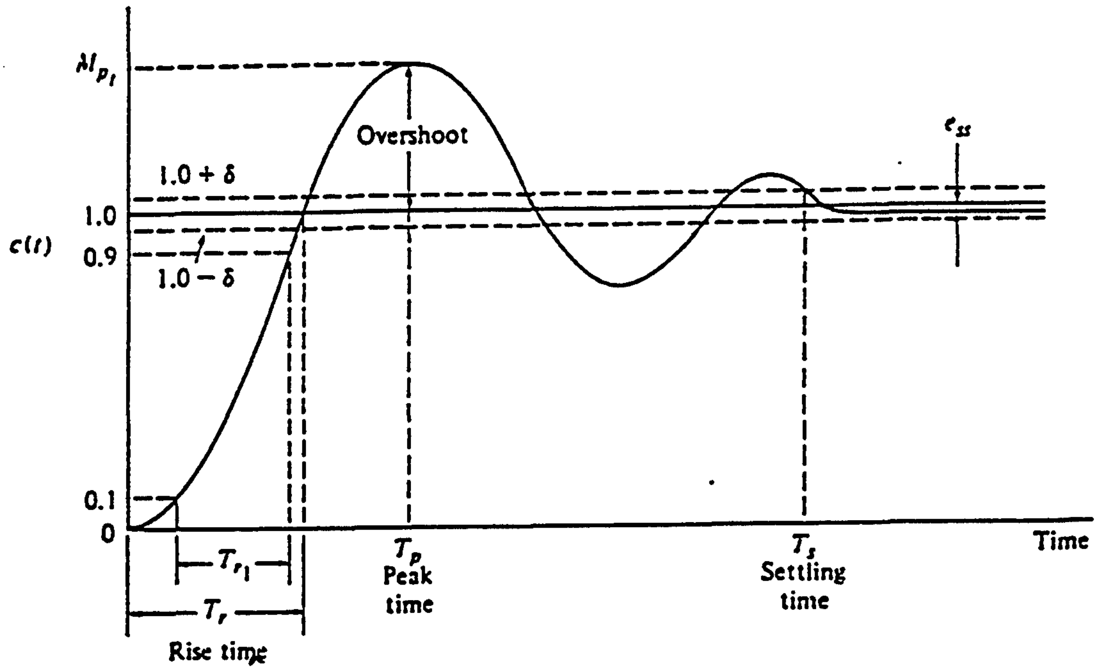


Figure 5-3. Step response of a control system.

The percent overshoot and peak time are plotted in Figure 5-4 versus the damping ratio ξ for a second-order system having the transfer function G(s) defined above. Experimental measurement of the percent overshoot allows the damping ratio to be evaluated using this figure. Measurement of the peak time then permits the natural frequency to be evaluated. When the damping factor and the natural frequency have been evaluated, a mathematical model having the form of G(s) can be constructed.

The curves presented in Figures 5-2 and 5-4 are exact only for a second-order system defined by the transfer function G(s). However, these figures also provide a good source of data for linear systems of higher order, because many higher-order systems possess a pair of dominant poles, i.e., a pair of poles much closer to the origin in the complex plane than any other poles of the system transfer function. For these higher-order systems, the step response can be estimated by means of the previous figures, and, conversely, a second-order approximate model can be identified based on experimental measurements of the unknown system's step response.

Frequency response experimental techniques can also be effectively combined with classical graphical design methods. The frequency response of a linear time-invariant system can be experimentally determined by applying a sine wave input of known amplitude, frequency and phase angle, waiting for the transient response to disappear, and recording the amplitude and relative phase shift of the output. This experiment is repeated for a number of different frequencies over the frequency range of interest, and the results presented in a Bode plot. Standard techniques from classical control theory can then be used to approximate the experimental Bode plot by a transfer function representation.

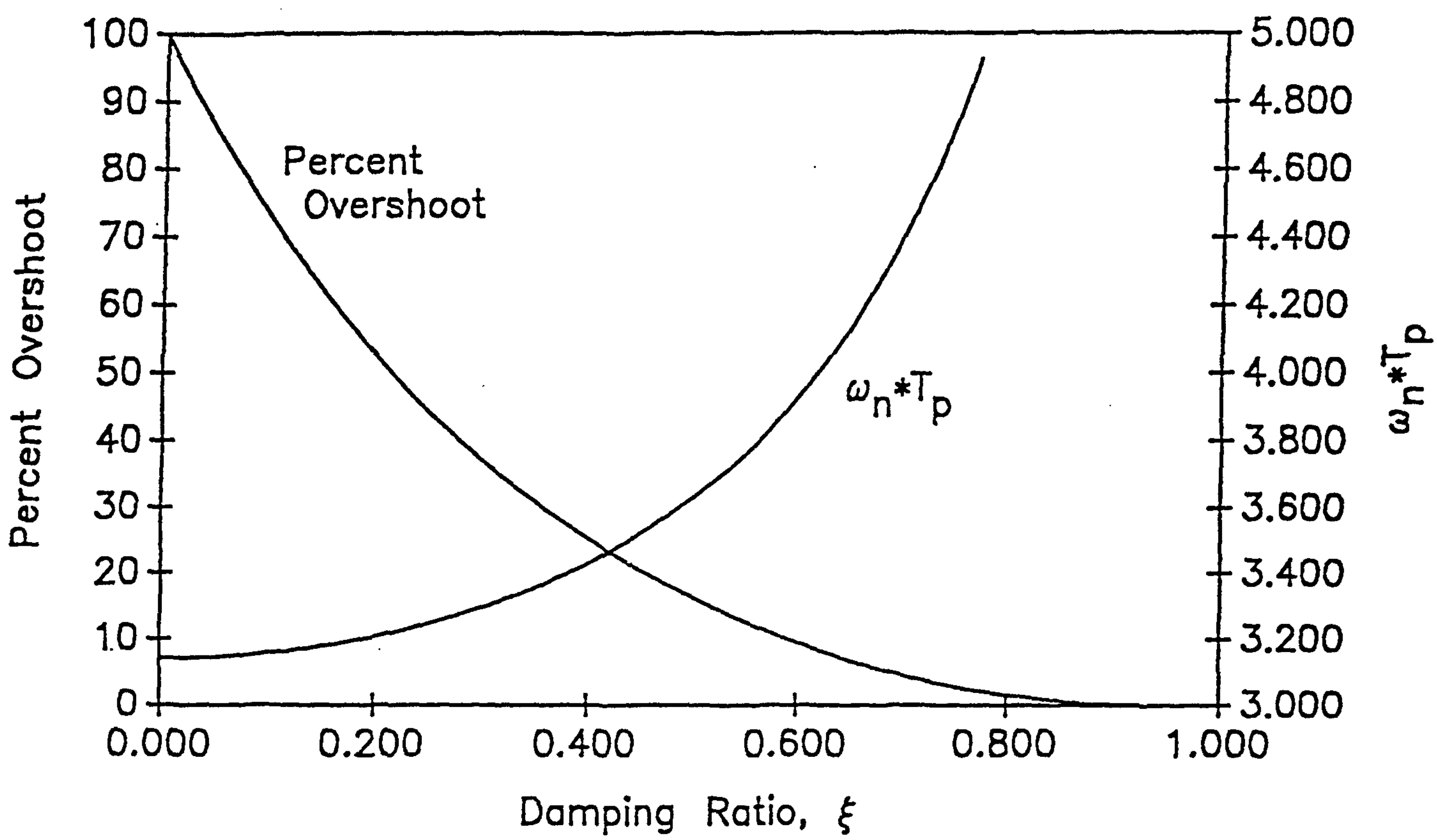


Figure 5-4. Percent overshoot and peak time versus damping ratio for a second-order dynamic system.

5.2 Identification Methods in Modern Control Theory

Modern control theory relies on analytical modeling methods for system identification. These techniques primarily involve time-domain measurements of the dynamic system's response and the automated use of mathematical model-fitting or optimization techniques.

System identification deals with the problem of developing mathematical models of dynamic systems using measured input and output data. In the time domain it is possible to continuously adjust the parameters of a selected system model so as to best fit the applied inputs and observed outputs. To apply this technique, a set of candidate models is selected and a criterion of fit between the model set and the observed data are chosen. That particular model which best describes the observed data according to the criterion of fit is selected to represent the unknown dynamic system.

Time-domain methods of system identification allow for a large number of different methods, model sets, and algorithms for computing goodness of fit. The system models are structured as predictors of the unknown dynamic system's output, and the identification criteria is based on a sequence of prediction errors.

Dynamic models used for system identification in the time domain may be in the form of linear difference equations, auto-regressive, moving-average, exogenous variable (ARMAX) time series processes, output error models, or multidimensional state variable models. In each case, a term is added to account for random noise sources and disturbances that affect the system and model inaccuracies. The noise sequences are usually assumed to be independent at different time instants and to have specified covariance matrices.

Criterion for determining the best model include the least-squares method, the maximum likelihood method, or other methods which depend on the model set and the goodness of fit criterion selected.

The basic concept for implementing these methods is to let each of the candidate mathematical models predict the next output y(t) based on the information available for all preceding time increments. The one candidate model which produces the best (minimal) sequence of errors between predictions and actual recorded outputs is selected as the best representation of the unknown dynamic system. The application of analytical modeling methods thus involves an optimization process conducted over the set of candidate mathematical models and the recorded sequence of input and output data.

The quality with which a particular mathematical model fits the observed data is crucial to the success of the system identification process. Given an observed data set and a specified mathematical model set, the best model is implicitly defined as the result of a numerical optimization process. Efficient, accurate numerical optimization algorithms are required to successfully implement this process.

If the mathematical model which results from the system identification process is required to operate on-line for purposes of adaptive control, self-tuning, or monitoring, computation time and memory requirements may restrict the way in which the predictions are computed and the model results evaluated. Recursive system identification methods are used in such cases. Applications which do not require on-line system identification may use other methods such as the maximum likelihood method.

Figure 5-5 shows a single-input, single-output, discrete-time, linear, time-invariant system. The system is defined by a transfer function T(z) whose parameters ϕ are assumed to be unknown. The dynamic system may consist of a continuous-time linear time-invariant system connected to a discrete-time input sequence u(k) by a digital-to-analog converter (DAC). The output of the continuous system is sampled by an analog-to-digital converter (ADC) and made available as a discrete-time output sequence y(k). The measured output sequence z(k) is assumed to be a noise-corrupted version of y(k):

$$z(k) = y(k) + w(k)$$

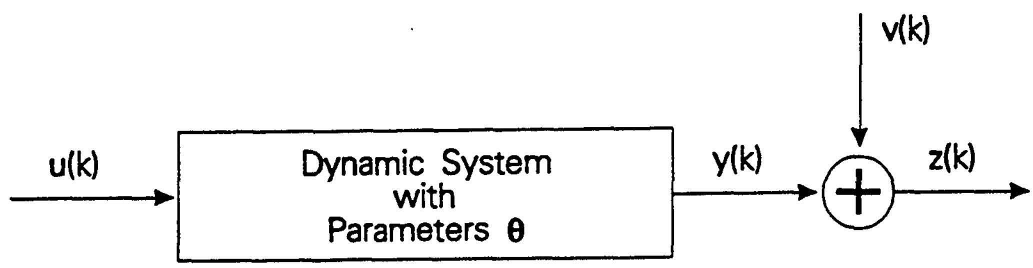


Figure 5-5. Single-input, single-output discrete linear time-invariant system.

where w(k) is a noise sequence which is usually, but not always, a white Gaussian noise process. The dynamic system is assumed, for the purposes of system identification, to be modeled by an auto-regressive, moving-average (ARMA) process represented by the following difference equation:

$$y(k) = \sum_{n=0}^{n=m} \theta_{nu}(k-n) + \sum_{n=1}^{n=2m} \theta_{n+m} y(k-n) .$$

The system parameters θ_j are assumed to be unknown constants. The purpose of the system identification procedure is to generate numerical estimates of the values of the 2m parameters θ_1, θ_2, ... θ_{2m} given a record of the input sequence u(k), the output sequence z(k), and some knowledge of the noise sequence w(k).

The measured output z(k) can be written as:

$$z(k) = \theta_0 u(k) + \theta_1 u(k-1) + \ldots + \theta_m u(k-m)$$

$$\theta_{m+1} y(k-1) + \theta_{m+2} y(k-2) \ldots + \theta_{2m} y(k-m) + w(k)$$

or in matrix form as:

$$z(k) = g^T(k)\theta + w(k)$$

where $g(k) = [u(k), u(k-1), \ldots, u(k-m), y(k-1), y(k-2), \ldots, y(k-m)]^T$

and $\theta = [\theta_1, \theta_2, \ldots, \theta_{2m}]^T$.

The vector g(k) contains all measured data available at time k.

The output equation for z(k) can be used to eliminate y(k) from the equation for z(k):

$$y(k) = (k) - w(k)$$

$$z(k) = \theta_0 u(k) + \theta_1 u(k-1) + \ldots + \theta_m u(k-m)$$

$$+ \theta_{m+1}\big(z(k-1)\, w(k-1)\big) + \theta_{m+2}\big(z(k-2) - w(k-2)\big)$$

$$+ \ldots + \theta_{2m}\big(z(k-m)\, w(k-m)\big) + w(k) .$$

Collecting terms, the matrix equation results:

$$z(k) = h^T(k)\theta + v(k),$$

where

$$h(k) = [u(k), u(k-1), \ldots u(k-m), z(k-1), z(k-2), \ldots, z(k-m)]^T$$

$$\theta = [\theta_0, \theta_1, \ldots, \theta_{2m}]^T,$$

and the noise process v(k) is:

$$v(k) = w(k) - \theta_{m+1} w(k-1) - \theta_{m+2} w(k-2) - \ldots - \theta_{2m} w(k-m) .$$

The discrete-time noise process v(k) is the result of passing the noise process w(k) through a linear system whose properties depend on the unknown parameters, and as a result, v(k) is generally non-white in nature.

The matrix equation:

$$z(k) = h^T(k)\, \theta + v(k)$$

applies at time k, and similar equations can be written for times $(k-1)$, $(k-2)$, ..., $(k-L)$, where L is the number of input/output data pairs to be used in the system identification process:

$$z(k) \quad = h^T \theta + v(k)$$

$$z(k-1) = h^T(k-1)\, \theta + v(k-1)$$

$$z(k-2) = h^T(k-2)\, \theta + v(k-2)$$

...

...

$$z(k-L) = h^T(k-L)\,\theta + v(k-L).$$

These scalar equations can be placed in matrix form to provide an overall measurement equation:

$$Z(k) = H(k)\,\theta + V(k),$$

where

$$Z(k) = [z(k),\ z(k-1),\ z(k-2),\ \ldots,\ z(k-L)]^T$$

is a column vector of noise and H(k) is an L by 2m matrix containing the prior L input/output data sets.

5.3 Recursive Methods of System Identification

Application of modern control theory to tactical weapon systems often requires a model of the underlying dynamic system to be available on-line, operating in real time in parallel with the actual dynamic system. The model may be needed for on-line decision purposes, for example: the choice of a suitable input signal during adaptive control, or the tuning of a filter by means of adaptive signal processing, monitoring, or fault detection. These on-line problems are amenable to solution by means of recursive system identification.

Recursive system identification means that the measured input and output data are processed sequentially in time as they occur and become available. Recursive system identification is also called real-time or on-line identification or sequential parameter estimation. This type of process is also referred to as an adaptive algorithm. The input and output pair at time k is denoted by:

$$z(k) = (u(k),\ y(k)),$$

and the parameter estimate at time k is denoted by $\theta'(k)$.

When performed off-line, an estimate of the parameter vector $\theta'(K)$ can be computed based on the complete collection of input and output data, as in the maximum likelihood method. Such batch processing methods cannot be used on-line, since the evaluation of $\theta'(t)$ may involve a large number of computations which may not terminate before the next sampling time. A recursive identification algorithm is thus required which has the following form:

$$x(k) = F[k,\ x(k-1),\ z(k)]\,,$$

$$\theta'(k) = f[x(k)]\,,$$

where x(k) is the information state, which contains all the data required to predict the next information state based on the present state and the effective input z(k). The functions F(.) and f(.) are expressions that can be evaluated with a known number of operations, and these operations can be completed before the next sampling time. By doing so, the system parameters $\theta'(k)$ can be evaluated during one sample time.

Recursive algorithms have been developed by many workers, each pursuing a different approach. Tsypkin[5.1] has applied a stochastic approximation method, based on the Robbins-Monroe algorithm[5.2].

The system identification problem can also be cast as a nonlinear state estimation or filtering problem by applying a Bayesian approach. The extended Kalman filter, as demonstrated by Ljung[5.3] is an example of this technique.

A third approach is the use of an adaptive observer, for example as in Luders and Narendra[5.4], and finally Ljung and Soderstrom[5.5] have presented a fourth approach which develops recursive algorithms based on existing off-line identification methods.

5.4 Least-Squares Methods

The least-squares method of system identification is the most commonly used time-domain method. The method is based on Gauss's well-known method of minimizing the sum of a sequence of squared terms. The least-squares method is also a standard mathematical tool for developing and computing statistical linear regression models.

The simplest model of a linear discrete-time system is the linear difference equation:

$$y(k) + a_1 y(k-1) + \ldots + a_n y(k-n) =$$

$$b_1 u(k-1) + \ldots + b_m u(k-m) + v(k)$$

where

$y(k)$ = output at time k
$u(k)$ = input at time k
$v(k)$ = errors and disturbances at time k.

The sources of v(k) are measurement errors, process disturbances and modeling errors. Let

$\theta = (a_1, \ldots, a_n, b_1, \ldots, b_m)^T$ and

$f(k) = (-y(k-1), -y(k-2), \ldots, -y(k-n),$

$u(k-1), u(k-2), \ldots, u(k-m))^T$.

The system can then be described in matrix form by:

$$y(k) = \theta^T f(k) + v(k) .$$

The least-squares identification problem involves finding suitable values for the lags n and m and the parameters θ based on observations of y(k) and u(k) for k = 1, 2, ..., N. The disturbance term v(k) is assumed to not be available.

The disturbance term v(k) is called the equation error and represents the numerical remnant that is not explained by the model structure. Given values for y(k) and f(k), this error can be determined as:

$$e(k) = y(k) - \theta^T f(k) .$$

Applying Gauss's method, one minimizes the sum of the squares of these errors:

$$\min_\theta V_N(\theta) = \left[\frac{1}{N}\right] \sum_{k=1}^{k=N} \left(y(k) - \theta^T f(k)\right)^2 .$$

The minimum value of V_N yields the least-squares estimate of the parameter vector θ and, in turn, identifies the parameters of the dynamic system model.

Up to this point we have illustrated how to identify the system when the structure, given by the lags n and m, of the dynamic system model has been specified. The choice of n and m is related to the desired complexity of the model and the acceptable goodness of fit. For any set of recorded data, a better fit will always be obtained by increasing n and m. One way to select n and m is to allow them to increase until the residual errors produced by the model are sufficiently small, and appear to be uncorrelated at different time instants.

5.5 Least-Squares System Identification

There are several variations to the least-squares identification method, including recursive least squares, weighted least squares, and multivariable models. The basic least squares method for system identification is summarized in Borrie[5.6].

The least squares system identification process provides a numerical estimate of the unknown parameters θ based on the input/output information available up to and including the present time k. To develop the least squares procedure, a dynamic system model, defined by a discrete-time transfer function T(z), is fed the same information, H(z), as the actual system. This process was illustrated in

Figure 2-10. The assumed system model generates an output $Z'(k)$. This output is compared with the actual system output $Z(k)$ and the error, $E(k)$, is calculated:

$$E(k) = Z(k) - Z'(k)$$

$$Z'(k) = H(k)\theta \ .$$

$Z'(k)$ is the output of the model based on the estimated numerical values of the system parameters at time k, and $Z(k)$ is the actual measured system output at time k.

The error $E(k)$ is then used as a feedback signal to drive a mathematical procedure which selects new parameter estimates, $\theta'(k)$, so that a performance measure $J(\theta'(k))$ is minimized. The performance measure used is the weighted sum of the squared errors over the L most recent input/output data pairs:

$$J\left(\theta'(k)\right) = E^{T}(k)\, W(k)\, E(k) \ .$$

In this performance measure, $W(k)$ is a positive definite weighting matrix, usually diagonal in form:

$$w(k) = \begin{bmatrix} w_0(k) & 0 & \dots & 0 \\ 0 & w_1(k) & \dots & 0 \\ . & . & \dots & 0 \\ 0 & 0 & \dots & w_L(k) \end{bmatrix}$$

By substituting the matrix equations for $E(k)$ and $Z'(k)$ into the performance measure, taking a derivative with respect to the parameter vector, and setting the result to zero, the weighted least-squares numerical estimate of the unknown parameters is obtained:

$$\theta'(k) = \left[H^{T}(k)\, w(k)\, H(k)\right]^{-1} \left[H^{T}(k)\, w(k)\, Z(k)\right] .$$

This is the best estimate of the unknown parameters θ based on the collection of input/output pairs available at times k, $k-1$, $k-2$, ..., $k-L$.

When the $(L+1)$ by $(L+1)$ weighting matrix $W(k)$ is selected as the $(L+1)$ identity matrix, the ordinary least-squares estimate $\theta'(k)$ is obtained:

$$\theta'(k) = \left[H^{T}(k)\, H(k)\right]^{-1} \left[H^{T}(k)\, H(k)\right] .$$

The elements of W(k) can also be selected to more heavily weight earlier or later measurements h(k). According to Mendel, this approach is useful when the unknown dynamic system possesses a transfer function whose parameters evolve slowly over time[5.7]. One strategy for time-weighting is to assign the elements of W(k) according to the rule:

$$w_i(k) = a^{k-L+i} .$$

Earlier measurements will then be weighted more heavily when $a > 1$, and later measurements will be weighted more heavily when $a < 1$.

As the amount of measured data (governed by the parameter L) is increased, the reliability of the numerical estimate $\theta'(k)$ generally improves. However, the computation of either the weighted or ordinary least-squares estimate requires the multiplication of several potentially large matrices and the inversion of an (L+1) by (L+1) matrix for each new estimate. Since these operations can be very time consuming, least-squares parameter estimates are generally performed by a recursive algorithm[5-6]:

$$\theta'(k+1) = \theta'(k) + P(k+1)\,h(k+1)\,w(k+1)\left[z(k+1) - h^T(k+1)\,\theta'(k)\right]$$

where

$\theta'(k+1)$	=	a 2m by 1 column vector of updated parameters,
$\theta'(k)$	=	a 2m by 1 column vector of prior parameter,
$P(k+1)$	=	an (L+1) by (L+1) matrix,
$h(k+1)$	=	a 2m by 1 column vector containing the information available at time (k+1),
$w(k+1)$	=	a scalar weighting factor applied to the information at time (k+1),
$z(k+1)$	=	the measured output at time (k+1).

The matrix P(k+1) is recursively computed using:

$$P(k+1) = \left[h(k+1)\,w(k+1)\,h^T(k+1)\,P^{-1}(k)\right]^{-1} .$$

This recursive procedure is initialized by accumulating (k+1) input/output data sets and computing an initial matrix P(k) as:

$$P(k) = \left[H^T(k)\,W(k)\,H(k)\right]^{-1} ,$$

where H(k) and W(k) are as previously defined. Alternatively, the process may be simply started[5.8] at time $k = 0$ with $P(k) = K$, an (L+1) by (L+1) diagonal matrix whose non-zero elements are set to a large positive number.

5.6 Least-Squares System Identification Numerical Example

As an example of the least-squares procedure for system identification, consider a single-input, single-output, time-invariant linear system. This system is illustrated in Figure 5-6.

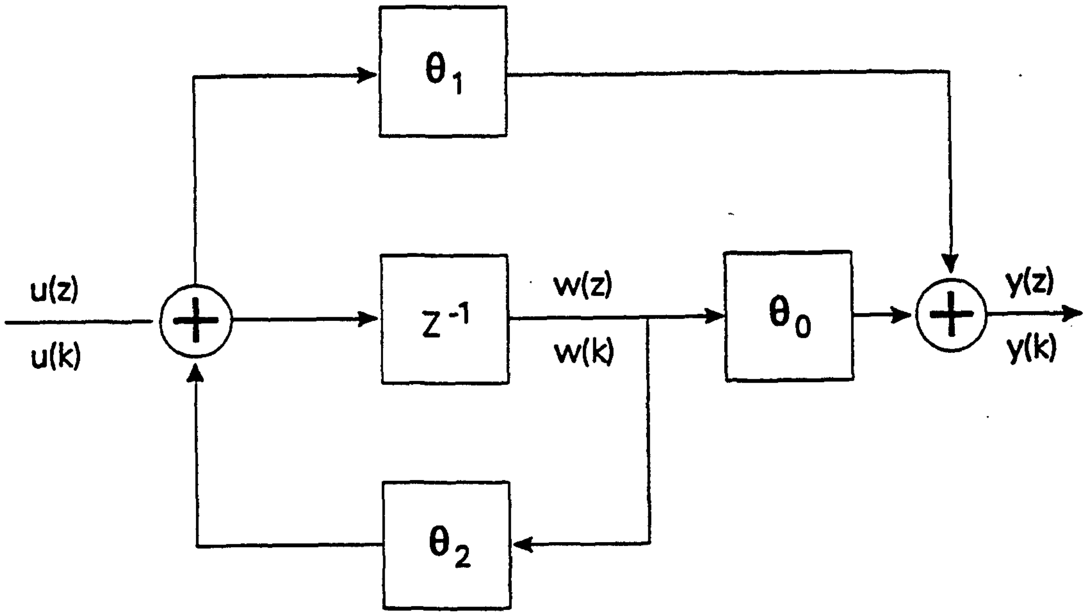


Figure 5-6. Single-input, single-output, time-invariant linear system.

The underlying dynamic system is assumed to have the following transfer function:

$$T(z) = \left[\frac{\theta_0 + \theta_1 z^{-1}}{1 - \theta_2 z^{-1}}\right] = \frac{Y(z)}{U(z)} .$$

From this transfer function, the following difference equation can be derived:

$$y(k) = \theta_0 u(k) + \theta_1 u(k-1) + \theta_2 y(k-1) .$$

Then,

$$z(k) = \theta_0 + \theta_1 u(k-1) + \theta_2 y(k-1) + w(k) ,$$

and applying the measurement equation,

$$z(k) = \theta_0 u(k) + \theta_1 u(k-1) + \theta_2 z(k-1) + v(k)$$

where

$$v(k) = w(k) - \theta_2 w(k-1) .$$

In matrix form this becomes:

$$z(k) = h^T(k)\,\theta(k) + v(k) \;,$$

where

$$h(k) = [u(k),\; u(k-1),\; z(k-1)]^T$$

and

$$\theta(k) = [\theta_0,\; \theta_1,\; \theta_2]^T \;.$$

To obtain a numerical estimate of the parameters θ, and thereby identify the structure of the dynamic system, an experiment was conducted. The input sequence:

$$u(k) = \{-1,\; 1,\; 2,\; 2,\; 0,\; -1,\; 0,\; 2,\; 2,\; \ldots\}$$

was applied to the unknown system and the output sequence:

$$z(k) = \{0,\; -0.63,\; 0.40,\; 1.41,\; 1.79,\; 0.66,\; -0.39,\; -0.14,\; 1.21,\; \ldots\}$$

was measured. In the assumed transfer function model, m = 1. A value of L = 2 was selected. The time at which the evaluation was performed was k = 4. At that time:

$$z(4) = \begin{bmatrix} z(4) \\ z(3) \\ z(2) \end{bmatrix} = \begin{bmatrix} 1.79 \\ 1.41 \\ 0.40 \end{bmatrix} ,$$

and

$$\begin{bmatrix} h^T(4) \\ h^T(3) \\ h^T(2) \end{bmatrix} = \begin{bmatrix} u(4) & u(3) & z(3) \\ u(3) & u(2) & z(2) \\ u(2) & u(1) & z(1) \end{bmatrix} = \begin{bmatrix} 0 & 2 & 1.41 \\ 2 & 2 & 0.40 \\ 2 & 2 & -0.63 \end{bmatrix} .$$

The parameter values were computed using the normal least-squares procedure:

$$\theta'(k) = [H^T(k)\,H(9)K(0)]^{-1}[H^T(k)\,Z(k)] \;,$$

This process resulted in the numerical values:

$$\theta'(4) = [0,\; -0.6337,\; 0.3667]^T \;.$$

To illustrate the recursive least-squares procedure, suppose that the time is (k+1) = 5 and L = 2 is again selected. The recursive procedure begins with (L+1) = 3 input/output measurements and, at k = 4,

$$P(4) = \left[\begin{bmatrix} 0 & 2 & 1.41 \\ 2 & 2 & 0.40 \\ 2 & 1 & -0.63 \end{bmatrix}^T \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 2 & 1.41 \\ 2 & 2 & 0.401 \\ 2 & 1 & -0.63 \end{bmatrix} \right]^{-1} ,$$

$$P(4) = \left[\begin{bmatrix} 0 & 2 & 1.41 \\ 2 & 2 & 0.40 \\ 2 & 1 & -0.63 \end{bmatrix}^T \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 2 & 1.41 \\ 2 & 2 & 0.401 \\ 2 & 1 & -0.63 \end{bmatrix} \right]^{-1} ,$$

$$\theta'(4) = [0,\ -0.6337,\ 0.3667]^T ,$$

$$\mathbf{h}(5) = [u(5),\ u(4),\ z(4)]^T$$

$$= [-1,\ 0,\ 1.79]^T .$$

Using w(5) = +1 as a weighting factor, the new parameter estimates are calculated from:

$$\theta'(5) = \theta'(4) + [0,\ +0.00045,\ +0.00135]^T$$

$$= [0,\ -0.6333,\ 0.3681]^T .$$

This process can be repeated indefinitely to improve the parameter estimates over time. This example was computed using an actual system having the parameters $\theta = [0, -0.632, 0.368]^T$ and without the presence of measurement noise. Figure 5-6 shows the structure of the identified system.

5.7 Maximum Likelihood Method

Maximum likelihood methods for dynamic system identification are of special importance because they are generally applicable to a variety of model structures. The resulting estimates of system parameters have good asymptotic properties, converging to final values in reasonable amounts of computation time. The maximum likelihood principle was first applied to single-input single-output auto-regressive moving-average exogenous variable (ARMAX) models by Astrom and Bohlin[5-9].

System identification, parameter estimation, and statistical inference all deal with the problem of extracting information from a set of noisy observations modeled by a set of random variables. The observations are contained in the random vector $y = (y_1, y_2, \ldots, y_n)$. The probability density function of y is assumed to have the form:

$$f(y_1, y_2, \ldots, y_n, \theta) = f(y, \theta) ,$$

which is equivalent to a probability distribution of y over a set A:

$$\text{Prob}\,(y \in A) = \int_{x \in A} f(\theta, x)\, dx .$$

θ is a d-dimensional vector of parameters which describes the properties of the observed variables y. These parameters are assumed to be unknown. The basic identification technique is to compute the vector θ by means of the observation y. This is done by constructing an estimator having the form $\theta'(y)$. If the observed value of y is y^*, then the resulting estimate for the parameters is $\theta'(y^*)$. Many forms for the estimator function are possible.

In 1912 Fisher[5-10] introduced the particular estimator called the maximum likelihood estimator, which maximizes the probability of the observed vector y. Fisher defined this estimator by recognizing that if the joint probability density function is $f(y, \theta)$, then the probability that the particular observation y^* should occur is proportional to $f(y^*, \theta)$.

The quantity $f(y^*, \theta)f$ is a deterministic function of the unknown parameters θ, once the numerical value of the vector y^* is specified. Fisher called this quantity the likelihood function, and it represents the likelihood that the observation y^* should indeed have occurred. A reasonable estimator of θ can then be obtained by selecting the unknown values of θ so that the probability of the observed event is as high as possible. This is the maximum likelihood estimator of the parameter vector θ:

$$\theta_{ML}(y^*) = \arg\max_{\theta} f(y, \theta) .$$

When applied to the problem of dynamic system identification, the process of maximum likelihood identification requires that the input and output sequences up to time k be recorded. The system model is thought of as a predictor function which predicts the output of the system, y(k), at time k based on the inputs and outputs up to time $k - 1$. A prediction error is defined as the difference between the predicted and observed outputs. The prediction error is usually assumed to be Gaussian with a zero mean and a time-dependent covariance matrix.

A likelihood function is next derived which depends on the time k, the unknown parameters of the model, θ, and the prediction errors up to time k, which in turn depend on the applied inputs, the assumed model structure, and the observed outputs. Maximizing this function yields a set of values for the unknown model parameters θ, and the unknown system is thus identified.

In modern control theory, system identification is applied primarily to obtain discrete time, linear, time-invariant models of dynamic systems. The discrete time nature of the problem arises from the use of a digital computer equipped with ADC and DAC converters operating at a sample time T to collect the input and output data. Figure 5-5 showed the basic setup of the system identification problem for a single-input, single-output linear time-invariant system described by a set of unknown parameters θ.

The system is subjected to an input sequence u(k) and produces an output sequence z(k) given by:

$$z(k) = y(k) + v(k) ,$$

where v(k) is usually (but not necessarily) a white Gaussian noise process, and the system behavior is described by an auto-regressive moving-average (ARMA) mathematical model:

$$\begin{aligned} y(k) = {} & u(k)\theta_0 + u(k-1)\theta_1 + \ldots + u(k-m)\theta_m \\ & + y(k-1)\theta_{m+1} + y(k-2)\theta_{m+2} + \ldots + y(k-m)\theta_{2m} \end{aligned}$$

The 2m unknown system parameters (θ_0, θ_1, ..., θ_{2m}) are assumed to be constants. The objective of the system identification procedure is to identify, determine, or estimate the numerical values of these parameters given the observed input and output sequences u(k) and z(k) and perhaps some knowledge of the properties of the noise sequence v(k).

The maximum likelihood method of system identification is based on a relatively simple procedure from statistical analysis. Consider a sequence of independent, identically distributed n-dimensional random vectors y(1), ..., y(N), where each y(i) is assumed to be modeled by a multivariate Gaussian probability density function having a mean vector μ and a covariance matrix σ. The likelihood function for this problem is:

$$L\left(y(1), \ldots, y(N), \mu, \sigma\right) = -\left[\frac{nN}{2}\right] \ln(2\pi) - \ln\left(\det(\sigma)\right) - \sum_{k=1}^{k=N} \left(y(k) - \mu\right)^T \sigma^{-1} \left(y(k) - \mu\right) .$$

Differentiating the likelihood function with respect to μ and σ yields the following two equations which must be solved for the maximum likelihood estimates μ' and σ' of the parameters μ and σ:

$$-\left[\frac{N}{2}\right](\sigma')^{-1} + \left[\frac{1}{2}\right]\sum_{k=1}^{k=N}\left(y(k) - \mu'\right)\left(y(k) - \mu'\right)^T \sigma^{-1} = 0 ,$$

$$\sum_{k=1}^{k=N}(\sigma')^{-1}\left(y(k) - \mu'\right) = 0 .$$

Solving these two simultaneous equations for the estimates μ and σ yields:

$$\mu' = \left[\frac{1}{2}\right]\sum_{k=1}^{k=N} y(k) ,$$

and

$$\sigma' = \left[\frac{1}{2}\right]\sum_{k=1}^{k=N}\left(y(k) - \mu'\right)\left(y(k) - \mu'\right)^T .$$

The equation for the vector μ' is simply the sample mean of the N vectors y(1), ..., y(N), and the equation for the matrix σ' is the sample covariance computed about the sample mean. Thus, by processing the accumulated data the initially unknown parameters μ and σ are estimated to be μ' and σ', and the statistical nature of the underlying process is identified.

When applied to the identification of linear, time-invariant dynamic systems, a mathematical model is assumed for the structure of the unknown dynamic system, and the model and the unknown system are supplied with the same sequence of input signals. The sequence of errors between the model output and the observed system output is treated as a sequence of Gaussian random variables having an unknown mean value and unknown variance. The observed sample mean and covariance matrix is a function of the model parameters. Analytic expressions are derived for the mean and covariance in terms of the model parameters, and the resulting nonlinear equations are solved numerically to optimize the model parameters in terms of the sample mean and covariance.

As an example, consider a generalized predictor model for a dynamic system having a single input u(k), an actual output y(k), and a predicted output Y(k) given by:

$$Y(k) = f\left(Y(k-1),\ u(k-1),\ \theta\right) + e(k) ,$$

where the sequence of prediction errors e(t) is assumed to be independent and identically distributed according to a probability density function $p_e(e\,|\,\theta)$. The likelihood function for this process is:

$$L\left(y(1),\ \ldots,\ y(N),\ \theta\right) = \ln\left(p_y\left(y(1),\ \ldots,\ y(N),\ \theta\right)\right)$$

$$= \ln\left[\prod_{k=1}^{k=N} P_y\left(y(k)\ |\ Y(k-1),\ u(k-1),\ \theta\right)\right]$$

$$= \sum_{k=1}^{k=N} \ln\left(P_y\left(y(k) \mid Y(k-1),\ u(k-1),\ \theta\right)\right)$$

$$= \sum_{k=1}^{k=N} \ln\left(P_e\left(y(k)\ -\ f\left(Y(k-1),\ u(k-1),\ \theta\right) \mid \theta\right)\right).$$

The prediction error $e(k) = y(k) - f(Y(k-1), u(k-1), \theta)$ appears in the likelihood function.

If the prediction errors e(k) are assumed to have a Gaussian probability density function with a mean of zero and a covariance matrix σ, the likelihood function can be written as:

$$L\left(y(1),\ \ldots,\ y(N),\ \theta\right) = \sum_{k=1}^{k=N} \left[-\left[\frac{n}{2}\right] \ln(2\pi) - \left[\frac{1}{2}\right] \ln\left(\det(\sigma)\right)\right]$$

$$-\left[\frac{1}{2}\right] \sum_{k=1}^{k=N} \left[\left(y(k) - f\left(Y(k-1),\ u(k-1),\ \theta\right)\right)^T \sigma^{-1} \left(y(k) - f\left(Y(k-1),\ u(k),\ \theta\right)\right)\right].$$

Differentiating this expression with respect to the unknown covariance matrix σ yields a solution for the maximum likelihood estimate, σ':

$$\sigma' = \left[\frac{1}{2}\right] \sum_{k=1}^{k=N} \left[\left(y(k) - f\left(Y(k-1),\ u(k-1),\ \theta\right)\right)\left(y(k) - f\left(Y(k-1_{,}\ u(k-1),\ \theta\right)\right)^T\right].$$

This equation for σ' is then used to eliminate σ from the likelihood function:

$$L\left(y(1),\ \ldots,\ y(N),\ \sigma\right) = -\left[\frac{nN}{2}\right] \ln(2\pi+1) - \left[\frac{1}{2}\right] \ln\left(\det(\sigma')\right).$$

For a single-input, single-output (scalar) dynamic system the likelihood function becomes:

$$L\left(y(1),\ \ldots,\ y(N),\ \sigma\right) = \text{Constant} - \left[\frac{N}{2}\right] \ln\left[\left[\frac{1}{N}\right] \sum_{k=1}^{k=N} \left(y(k) - f\left(Y(k-1),\ u(k-1),\ \theta\right)\right)\right]^2.$$

Maximization of the likelihood function is then achieved by minimizing the logarithm of the prediction error covariance. The numerical value of the parameter θ is determined by a numerical optimization process. Note that the function f(.) determines the relationship between the model input u(k), model output Y(k), and parameter θ.

If a linear scalar model is assumed for the unknown system,

$Y(k) = f\left(Y(k-1),\ u(k-1),\ \theta\right) + e(k)$ becomes

$$Y(k) = \theta_1\ Y(k-1) + \theta_2 U(k-1) + e(k),\ k = 1,\ 2,\ \ldots,\ N.$$

The values of the measurement error e(k) are assumed to be drawn from a normal distribution having a mean value of zero and a variance of σ^2.

A sequence of N input signals is then applied to the model and the unknown system and the model and system outputs, Y(k) and y(k), are recorded along with the inputs u(k). The likelihood function for this problem becomes:

$$L\big(y(1), \ldots, y(N), \theta\big) = \text{Constant} - \left[\frac{N}{2}\right] \ln^2\left[\left[\frac{1}{N}\right] \sum_{k=1}^{k=N} \big(y(k) - f\big(Y(k-1), u(k), \theta\big)\big)\right]$$

$$= \text{Constant} - \left[\frac{N}{2}\right] \ln^2\left[\left[\frac{1}{N}\right] \sum_{k=1}^{k=N} \big(y(k) - \theta_1 Y(k-1 - \theta_2 u(k-1))\big)\right] .$$

The likelihood function will be maximized if the expression:

$$\ln^2\left[\left[\frac{1}{N}\right] \sum_{k=1}^{k=N} \big(y(k) - \theta_1 Y(k-1) - \theta_2 u(k-1)\big)\right]$$

is minimized. In general, numerical optimization techniques similar to those discussed later in this review must be used to solve for the best values of the unknown parameter values θ_1 and θ_2.

Franklin and Powell[5.11] discuss the numerical implementation of the maximum likelihood method for system identification and present an algorithm suitable for off-line or on-line operation.

5.8 Maximum Likelihood Estimation of System Parameters

The maximum likelihood method is an off-line, batch-type computation process also applied to the collection of input/output information available at time k:

$$Z(k) = H^T(k)\theta + V(k)$$

where

$$Z(k) = \big[z(k), z(k-1), \ldots, z(k-L)\big]^T$$

is an L by 1 column vector,

$$H(k) = \begin{bmatrix} h^T(k) \\ h^T(k-1) \\ h^T(k-L) \end{bmatrix} = \begin{bmatrix} u(k) & u(k-1) & \ldots & u(k-m) & z(k-1) & z(k-2) & \ldots & z(k-m) \\ u(k-1) & u(k-2) & \ldots & u(k-1-m) & z(k-2) & z(k-3) & \ldots & z(k-1-m) \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ u(k-L) & u(k-1-L) & \ldots & \ldots & \ldots & \ldots & \ldots & z(k-L-m) \end{bmatrix}$$

is an L by 2m matrix containing the prior L input/output data sets, and

$$\theta = [\theta_0, \theta_1, \ldots, \theta_{2m}]^T$$

is a (2m+1) by 1 column vector containing the unknown system parameters.

In the maximum likelihood method, the noise vector V(k) is assumed to represent a zero mean Gaussian process. The (L+1) Gaussian joint probability density function for the measured output Z(k) is given as[5-6]:

$$p(Z(k)) = \left[(2\pi)^{L+1} \det(R(k))\right]^{-1/2} \exp\left\{-(1/2)\left[Z(k) - H(k)\theta'(k)\right]^T R^{-1}(k)\left[Z(k) - H(k)\theta'(k)\right]\right\} .$$

where the (L+1) by (L+1) matrix R(k) is the expected value of the matrix product $V(k)V^T(k)$. R(k) is called the covariance matrix of the discrete noise sequence.

The values of the estimated parameters $\theta'(k)$ are selected by the maximum likelihood method so that p(Z(k)) is maximized for the available observations Z(k) and H(k). By maximizing p(Z(k)), the observations Z(k) are considered to be as likely as possible. Maximization of p(Z(k)) is equivalent to minimizing the likelihood function L(Z(k)),
where:

$$L(Z(k)) = \left[Z(k) - H(k)\theta'(k)\right]^T R^{-1}(k)\left[Z(k) - H(k)\theta'(k)\right] .$$

The maximum likelihood estimate of $\theta'(k)$ is obtained by taking a derivative of the likelihood function, setting the result to zero and solving for the result:

$$\theta'(k) = \left[H^T(k) R^{-1}(k) H(k)\right]^{-1} H^T(k) R^{-1}(k) Z(k) .$$

To apply the maximum likelihood method, the noise characteristics of V(k), determined by the covariance matrix R(k), must be known. The resulting estimate, $\theta'(k)$, is bias-free.

In most applications, no information regarding the statistical properties of V(k) is available. However, the maximum likelihood method can be extended to identify the parameters of the noise process as well as those of the unknown dynamic system. The approach below can also be applied to the least-squares method, and is related to Johnson's disturbance accommodating control method described by Borrie[5,6].

As a technique for identifying the noise process as well as the dynamic system, Saridis[5,6] recommends that the measurement process be written as:

$$z(k) = a_0 u(k) + a_1 u(k-1) + \ldots + a_m u(k-m)$$
$$+ b_1 z(k-1) + b_2 z(k-2) + \ldots + b_m z(k-m)$$

$$+ c_0 w(k) + c_1 w(k-1) + \ldots + c_m w(k-m) .$$

The noise process w(k) is assumed to be white, Gaussian, and to have a mean of zero and an unknown variance.

A maximum likelihood estimate, z′(k), of the output z(k) is assumed, based on the available evidence contained in the input/output information present at time k. The parameters a_j, b_j and c_j are assumed to be estimated as a'_j, b'_j and c'_j. An error equation is then written:

$$e(k) = z(k) - z'(k)$$

$$e(k) = z(k) - \sum_{j=0}^{j=m} a_j' u(k-j) - \sum_{j=1}^{j=m} b_j' z(k-j) - \sum_{j=0}^{j=m} c_j' e(k-j) .$$

The estimated parameter vector $\theta'(k)$, a (3m+2) by 1 column vector, is written as:

$$\theta'(k) = \left[a_0', a_1', \ldots, a_m', b_1', b_2', \ldots, b_m', c_0', c_1', \ldots, c_m'\right]^T .$$

M + m sets of input/output data are then accumulated, and a probability density function for the error is set up. The logarithm of that density function serves as a likelihood function:

$$L(\theta', \sigma^2) = \text{Constant} - \left[\frac{M}{2}\right] \ln(\sigma^2) - \left[\frac{1}{2\sigma^2}\right] \sum_{k=m+1}^{k=m+M} e^2(k) .$$

In this expression, σ^2 is the unknown variance of the error e(k). Maximizing $L(\theta', \sigma^2)$ is equivalent to minimizing:

$$J(\theta') = \sum_{k=m+1}^{k=m+M} e^2(k) .$$

When the minimum is attained, the error variance σ^2 becomes:

$$\sigma^2 = \left[\frac{1}{M}\right] J(\theta') .$$

5.9 A Steepest Descent Algorithm for Maximum Likelihood Parameter Estimation

Borrie[5-6] suggests that $J(\theta')$ be minimized by means of a steepest descent numerical method. Either the same block of input/output information can be used at each iteration of that algorithm, or a new block of M input/output information can be assembled and used at each successive iteration. The structure of this algorithm applied to maximum likelihood estimation of system parameters and the identification of a discrete-time dynamic system is outlined below.

Step 1. Assume sets of initial values for the unknown parameters and the error gradient:

$$\theta'(k) = \left[a_0', a_1', \ldots, a_m', b_1', b_2', \ldots, b_m', c_0', c_1', \ldots, c_m'\right]^T .$$

$$\frac{\partial e(k-j)}{\partial \theta'} = \left[\frac{\partial e(k-j)}{\partial a_0'}, \ldots, \frac{\partial e(k-j)}{\partial c_m'}\right]^T, j = 0, 1, \ldots, m .$$

Step 2. Evaluate the errors e(k), k = m + 1, m + 2, ..., m + M using:

$$e(k) = z(k) - \sum_{j=0}^{j=m} a_j' u(k-j) - \sum_{j=1}^{j=m} b_j' z(k-j) - \sum_{j=0}^{j=m} c_j' e(k-j) .$$

Step 3. Using the most recent parameter estimates θ' or θ'_{new}, evaluate the following partial derivatives for k = m + 1, m + 2, ..., (m+M):

$$\frac{\partial e(k)}{\partial a_j'} = -u(k-j) - \sum_{j=0}^{j=m} c_j' \frac{\partial e(k-j)}{\partial a_j'}, j = 0, 1, .., m$$

$$\frac{\partial e(k)}{\partial b_j'} = -z(k-j) - \sum_{j=0}^{j=m} c_j' \frac{\partial e(k-j)}{\partial b_j'}, j = 2, 2, .., m$$

$$\frac{\partial e(k)}{\partial c_j'} = -e(k-k) - \sum_{j=0}^{j=m} c_j' \frac{\partial e(k-j)}{\partial c_j'}, j = 0, 1, .., m$$

Step 4. Evaluate the rate of change of the performance measure:

$$\frac{\partial J}{\partial \theta'} = -2 \sum_{k=m+1}^{k=m+M} e(k) \frac{\partial e(k)}{\partial \theta'} .$$

Step 5. Evaluate the scalar S, where:

$$S \approx \frac{\partial^2 J}{\partial \theta^2} = \sum_{k=m+1}^{k=m+M} \left\{ \left[\frac{\partial e(k)}{\partial \theta'}\right] \left[\frac{\partial e(k)}{\partial \theta'}\right]^T \right\} .$$

Step 6. Revise the estimate of the parameters according to:

$$\theta'_{new} = \theta' - S^{-1} \frac{\partial J}{\partial \theta'} .$$

Step 7. Repeat the above procedure, beginning at Step 2, using the most recent parameter estimates θ_{new} and partial derivatives $\partial e(k-j)/\partial\theta'$, k = (m+M), j = 0, ..., m, in place of the Step 1 estimates. Terminate the algorithm when no further change in the parameter estimates is noted.

The maximum likelihood algorithm presented above can be expected to converge successfully if the initial parameter estimates are reasonably correct. A least-squares method might be used to provide the necessary initial estimates.

5.10 Summary

The problem of dynamic system identification, the development of a mathematical model for a dynamic system based on measurement of the system's input and output, has been discussed. Several methods of system identification, including the methods of least squares and maximum likelihood, were described in some detail.

System identification is closely related to the problem of state variable estimation and use of a Kalman filter which is discussed in the next chapter of this report. While the identification process attempts to estimate the parameters of a selected mathematical model for a potentially unknown dynamic system, the process of state variable estimation attempts to produce an estimate of the state variables present in a specific mathematical model. State variable estimation is necessary for implementing feedback in certain closed-loop control systems. System identification will later be seen to play a major role in the implementation of adaptive control systems, which may also employ state variable feedback.

Eyhoff[5.12] has developed and outlined a variety of system identification methods based on the least-squares method, the maximum likelihood method, and other techniques. Additional methods proposed included the use of extended Kalman filter algorithms and the statistical analysis of input and output data.

It should also be noted that a careful specification of the experimental input sequence is required to achieve reliable estimates of system parameters. These parameters, and the order of the unknown system, cannot be reliably identified using arbitrary input signals.

REFERENCES:

5.1 Tsypkin, Y.Z., Adaptation and Learning in Automatic Systems, Academic Press, New York, 1971.

5.2 Robbins, H. and Monroe, S., A Stochastic Approximation Method, Annals of Mathematical Statistics, Vol.22, pp.400-407, 1951.

5.3 Ljung, L., Asymptotic Behavior of the Extended Kalman Filter as a Parameter Estimator for Linear Systems, IEEE Transactions on Automatic Control, Vol.24, pp.36-50, 1979.

5.4 Luders, G. and Narendra, K.S., Stable Adaptive Schemes for State Estimation and Identification of Linear Systems, IEEE Transactions on Automatic Control, Vol.19, pp.841-847, 1974.

5.5 Ljung, L. and Soderstrom, T., Theory and Practice of Recursive Identification, MTI Press, Cambridge, MA, 1983.

5.6 Borrie, J.A., Modern Control Systems - A Manual of Design Methods, Prentice-Hall International, Englewood Cliffs, New Jersey, 1986.

5.7 Mendel, J.M., Discrete Techniques of Parameter Estimation, Marcel Dekker, New York, 1973.

5.8 Astrom, K.J. and Wittenmark, B.,Adaptive Control, Addison-Wesley Publishing Company, Reading, MA, 1989.

5.9 Astrom, K.J. and Bohlin, T., Numerical Identification of Linear Dynamic Systems from Normal Operating Records, Theory of Self-Adaptive Control Systems, Plenum Publishing Company, New York, 1965.

5.10 Fisher, R.A., On An Absolute Criterion for Fitting Frequency Curves, Meas. Math., Vol.41, P.155, 1912.

5.11 Franklin, G.F., and Powell, J.D., Digital Control of Dynamic Systems, Addison-Wesley Publishing Company, Reading, Massachusetts, 1980.

5.12 Eyhoff, P., Trends and Progress in System Identification, Pergammon Press, New York, 1981.

This page is intentionally blank.

CHAPTER 6
THE KALMAN FILTER

6.1 The Basic Estimation Problem

In modern control theory, the process of system identification, as outlined in the previous Chapter, is used to determine the structure of and parameter values for a dynamic system's mathematical model. These results are based on and computed using measurements of the system's input and output. System identification problems are mainly treated as deterministic problems, although mathematical methods borrowed from statistics are employed. In modern control theory, the term estimation usually refers to methods for determining the values of a dynamic system's state variables. When the parameters of a dynamic system are considered to be random variables, there is considerable mathematical overlap between the processes of system identification and the determination of the nature of a specific signal, and the evaluation of a specific parameter value. Estimation problems in modern control theory are largely stochastic in nature.

Estimation problems are an application of statistical decision theory. The basic estimation problem has the following five components:

(1) A set of unknown variables or parameters, Θ. The elements of Θ are most often the real-valued states of a dynamic system. Θ may also represent a vector of m uncertain, statistically variable parameters in a mathematical model of a dynamic system.

(2) A family of statistical probability distribution functions, F_Θ, indexed over the elements of Θ. These distribution functions may be either continuous or discrete.

(3) A scalar or vector random variable Y, whose statistical distribution $F_\Theta(Y)$ is a member of the family F_Θ, but with the corresponding Θ assumed to be unknown.

(4) A set of estimator functions $\Theta'(Y)$ which provide statistical estimates for the numerical values of the unknown parameters Θ based on the observation Y. Since Y is a random variable, so is the estimate $\Theta'(Y)$.

(5) A loss function $L(\Theta, \Theta'(Y))$ which represents the cost incurred by using the estimate $\Theta'(Y)$ instead of the true parameter value Θ. For a scalar parameter Θ, some common loss functions are the absolute error loss function:

$$L(\theta,\ \theta'(Y)) = \mathrm{abs}(\theta-\theta'(Y))\ ,$$

the square error loss function:

$$L(\theta,\ \theta'(Y)) = (\theta-\theta'(Y))^2\ ,$$

and the threshold loss function:

$$\begin{aligned} L(\theta,\ \theta'(Y)) &= 0,\ \mathrm{abs}(\theta-\theta(Y)) < \epsilon\ , \\ &= 1,\ \mathrm{abs}(\theta-\theta'(Y)) \geq \epsilon\ . \end{aligned}$$

The basic parameter estimation problem requires one to find an estimator $\Theta'(Y)$ such that the chosen loss function will be as small as possible. As an example, suppose that an observation Y has the following probability density function:

$$F_\theta(Y) = (2\pi)^{-1/2} \exp\left[\frac{-(Y-\theta)^2}{2}\right]\ .$$

The scalar observation Y is a Gaussian, or normal, random variable with a variance of 1 and an unknown mean value Θ. The problem is to estimate the unknown parameter Θ, the mean value of the distribution of Y, based on the observation of Y. Except for the unknown parameter value, the statistical distribution of Y is completely specified in this problem.

There are two fundamentally different approaches to solving parameter estimation problems. The first approach is the Bayesian approach in which the parameter Θ is itself considered to be a random variable. The form of the statistical distribution of Θ, $G(\Theta)$, is assumed to be known. This distribution is called the prior distribution of Θ. Observations of Y are then taken and the observed distribution of Y, $F_\Theta(Y)$ is then assumed to be a conditional distribution, the distribution of Y given a value of Θ. This conditional distribution is denoted by $F(Y|\Theta)$.

The second approach, the non-Bayesian approach, assumes that the unknown parameter Θ is constant. The non-Bayesian approach is mainly encountered in statistical applications, while the Bayesian approach has found wide acceptance in engineering applications. These engineering applications of parameter estimation are generally concerned with man-made observations or signals for which reasonable estimates of the prior distribution can be made. On the other hand, statistical applications normally involve naturally generated observations and for these signals it is usually impossible to determine prior distributions.

6.1.1 Bayesian Parameter Estimation

In the Bayesian approach the unknown parameter Θ is considered to be a random variable with an assumed probability density function G(Θ) and a corresponding probability density function g(Θ). An optimality criteria is then introduced. A loss function L(Θ, Θ′(Y)) is assumed to be given as part of the estimation problem's specification. Holding the value of Θ temporarily fixed, the loss function can be averaged over all possible outcomes for Y, yielding the risk function R(Θ, Θ′(Y)):

$$R\left(\theta,\ \theta'(Y)\right) = E\left[L\left(\theta,\ \theta'(Y) : \theta\right)\right]$$

$$= \text{INT}\ L\left(\theta,\ \theta'(Y)\right)\ dF\left(Y\ |\ \theta\right)\ .$$

The risk is a function of the parameter Θ, and a function of the estimate Θ′(Y), but is not a function of the observation Y. An example will help to clarify this process. Let the distribution of the parameter Θ be represented by a normal distribution with a mean value of zero and a variance σ^2:

$$g\left(\theta\right) = (2\pi)^{-1/2} \exp \frac{-\theta^2}{2\sigma^2}$$

Let the estimate Θ′(Y) equal cY, where c is a constant to be determined in an optimal manner. The true probability density function for Y is normal with a mean value of Θ and a variance of one:

$$f\left(Y\ |\ \theta\right) = (2\pi)^{-1/2}\ \exp \frac{-(Y-\theta)^2}{2}$$

The loss function is taken as the squared error, and the resulting risk function is:

$$R\left(\theta,\ \theta'(Y)\right) = \text{INT}\ (\theta - cY)^2\,(2\pi)^{-1/2}\ \exp \frac{-(Y-\theta)^2}{2\,dY}$$

$$= c^2 + (1-c)^2\ \theta^2\ .$$

The risk function is an explicit function of the unknown parameter Θ, and is functionally dependent on the estimate Θ′(Y) by means of the constant c.

Since Θ is in fact a random variable, the risk function can be further averaged over the distribution for Θ:

$$r\left(\theta'(Y)\right) = E\left[R\left(\theta,\ \theta'(Y)\right)\right] = \text{INT}\ R\left(\theta,\ \theta'(Y)\right) dG\left(\theta\right)\ .$$

The result, r(Θ′(Y)), is called the Bayes' risk associated with the use of the estimator Θ′(Y). In this example,

$$r\left(\theta'(Y)\right) = c^2 + (1-c)^2\,\sigma^2\ .$$

The optimal Bayes estimate for Θ is given by that estimate $\Theta'(Y)$ which minimizes the Bayes risk. By differentiating the Bayes risk function with respect to the constant c and equating the resulting expression to zero we obtain:

$$c = \frac{\sigma^2}{1 + \sigma^2}$$

and

$$\theta'(Y) = cY \ .$$

The previously unknown constant c has now been selected in an optimal manner. Recall that the probability distribution function for Θ was originally assumed to be known as $g(\Theta)$, so the variance σ^2 is also assumed to be known. By selecting other loss functions other estimators for the unknown parameter Θ can be derived from Sage and Melsa[6.1].

6.1.2 Non-Bayesian Parameter Estimation

The most important non-Bayesian estimator is the maximum likelihood estimator, defined implicitly by:

$$\theta_{ml}(Y) = \arg\left[\max_{\theta} f_{\theta}(Y)\right] .$$

The meaning of this expression is that the value of Θ which maximizes $f_{\Theta}(Y)$, the probability density function for Y, indicating that the observation Y was indeed most likely to occur, is accepted as the estimate of the unknown parameter Θ.

As an example, let $f_{\Theta}(Y) = (2\pi)^{-1/2} \exp(-(Y - \Theta)^2/2)$, then:

$$\begin{aligned} \theta_{ml}(Y) &= \arg\left[\max_{\theta} f_{\theta}(Y)\right] , \\ &= \arg\left[\max_{\theta} \left\{(2\pi)^{-1/2} \exp\left(-(Y-\theta)^2/2\right)\right\}\right] \\ &= Y \ . \end{aligned}$$

For this estimated value of Θ, $\exp(0) = 1$, and the probability density function is maximized.

The probability density function $f_{\Theta}(Y)$ is called the likelihood function. All statistical inferences regarding the parameter Θ should, according to the likelihood principle, be based only on analysis of the likelihood function.

6.2 Nonlinear Estimation

The nonlinear estimation problem involves the Bayesian estimation of a stochastic process x(t), usually the state of a dynamic system. Most often, this state cannot be observed directly. Information regarding the random process x(t) is obtained from a related process y(t), called the observation process. The goal in applications of nonlinear estimation is to compute, for each time t, a least-squares estimate of some nonlinear function of x(t) given the observation history up to y(t). One possible function of interest is the conditional distribution of x(t) given the observation y(t).

When the observations are received sequentially it is preferable to compute this estimate in a recursive manner, with the latest estimate updated based only on the most recently received observation. This leads to the design of nonlinear estimators which are memoryless in the sense that it is unnecessary to remember the entire observation history.

For the special case of a linear system with linear observations and additive Gaussian white noise this estimation problem was initially solved by Kalman and Bucy[6.2]. Their result, now generally called the Kalman filter, is a widely applied tool of modern control theory. Attempts have since been made to generalize their results to much more difficult problems involving both nonlinear system dynamics and nonlinear observations. Results for a variety of special cases and particular applications are available[6.3, 6.4]. In the next Section we provide an overview of the Kalman filter and selected applications.

6.3 Kalman Filtering and Applications

Kalman filtering is a technique for the recursive estimation of the state variables of a dynamic system based on a set of noisy measurements. The Kalman filter estimates the state of a dynamic system by combining in an optimal manner a knowledge of the system model, a set of a priori state estimates and the measurement noise characteristics. In contrast to classical least squares methods which require simultaneous processing of a large amount of measured data, the Kalman filter operates on the observed data in a sequential, recursive manner eliminating any requirement to store the entire measurement history.

The Kalman filter algorithm also produces an estimation error covariance matrix representing the uncertainty in the state estimates. This matrix does not depend on actual observed data for a linear dynamic system, making it possible to pre-calculate the covariance matrix for different models and measurement techniques. The information contained in the covariance matrix can be used to evaluate accuracy improvements resulting from additional or alternate sensors and the accuracy effects of additional state variables.

Kalman filter applications have been widely documented in textbooks, survey papers and journal articles[6.5, 6.6]. Kalman filters have been applied in such diverse applications as spacecraft orbit determination, satellite tracking, navigation, digital image processing, economic forecasting, industrial process control, power systems, and on-line fault detection.

6.3.1 Kalman Filter Algorithm

The state of a stochastic system can be estimated by means of a Kalman filter which uses the system input and output as data. The process is similar to the use of a Luenberger observer for a deterministic system. Observers are discussed in Section 3.10 of this report. Kalman filters now exist in several forms. The basic optimal form applies to linear time-varying systems. Suboptimal forms of the basic design apply to linear time-invariant systems and extended forms have been applied to certain classes of nonlinear systems. Figure 6-1 shows a discrete-time, time-variant stochastic system with an attached Kalman filter which generates an estimate $x_e(k)$ of the system state $x(k)$.

If the state of the dynamic system is completely observable, all of the system states can be estimated by means of a Kalman filter. The state transition and output equations of the dynamic system are:

$$x(k+1) = A(k)\,x(k) + B(k)\,u(k) + w(k)$$

$$y(k) = C(k)\,x(k) + D(k)\,u(k) + v(k)\,.$$

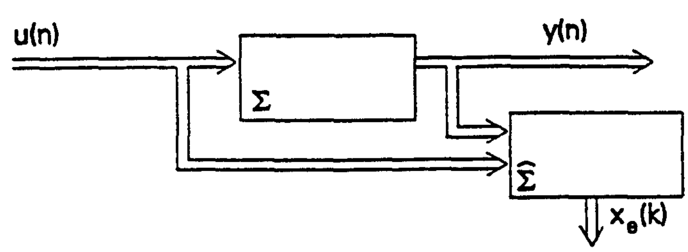


Figure 6-1. Discrete-time, time variant stochastic system with a Kalman filter.

In this mathematical model, $x(k)$ is a vector of n state variables, $u(k)$ is a vector of m input variables at time k, $y(k)$ is a vector of p output variables, $A(k)$ is an n by n time-varying matrix, $B(k)$ is an n by m time-varying matrix, $C(k)$ is a p by n time-varying matrix, and $D(k)$ is a p by m time-varying matrix.

The measurement noise processes are modeled by the n-dimensional vector w(k) and the p-dimensional vector v(k), both Gaussian white noise vectors whose properties are stationary, or time-invariant:

$$E\left[w(k)\right] = 0,\ \mathrm{cov}\left[w(k)\right] = Q\,\delta(k)$$

$$E\left[v(k)\right] = 0,\ \mathrm{cov}\left[v(k)\right] = R\,\delta(k)\ .$$

where Q and R are n by n matrices of constants.

The basic Kalman filter algorithm[6,7] consists of the following sequence of steps:

(0) Set $k = 0$
Input A(k), B(k), C(k), D(k), Q, R, G(0), $x_e(0)$
Set $k = k + 1$

(1) Compute $P(k) = R + C(k)G(k)C^T(k)$ and $P^{-1}(k)$,

(2) Compute $M(k) = A(k)G(k)C^T(k)P^{-1}(k)$,

(3) Compute $x_e(k + 1)=$
$A(k)x_e(k) + M(k)[y(k) - C(k)x_e(k) - D(k)u(k)] + B(k)u(K)$,

(4) Compute $G(k + 1) = [A(k) - M(k)C(k)]G(k)A^T(k) + Q$,

(5) Set $k = k + 1$
Go to Step (1).

The Kalman filter is an asymptotic state estimator. Starting with an initial guess of the state, $x_e(0)$, the Kalman filter determines the matrix M(k) which minimizes the performance measure:

$$\mathrm{trace}\left[G(k)\right] = E\left[\sum_{i=1}^{n} \left(x_i(k)-x_{je}(k)\right)^2\right]\ .$$

The matrix G(k) is the expected value of the error covariance matrix:

$$G(k) = E\left[\left(x(k)-x_e(k)\right)\left(x(k)-x_e(k)\right)^T\right]\ .$$

The successful application of the Kalman filter requires care in modeling the dynamic system. Poor numerical conditioning of the error covariance matrix G(k) can lead to unacceptable results.

The filter algorithm automatically positions the eigenvalues of the matrix $[A(k)-M(k)C(k)]$ so that the sum of the variances of the estimation errors, trace(G(k)), is minimized. If the underlying dynamic system is stable, the filter output $x_e(k)$ settles down, after an initial transient, to a behavior which is independent of the initial values G(0) and $x_e(0)$. In practice, the elements of these initial conditions can be set to arbitrary values, typically zero. If any information is available which permits the initial conditions to be set more accurately, the duration of the filter transient can be substantially reduced.

The Kalman filter gain, M(k), depends on the matrices Q and R which are measures of the amplitudes of the noise affecting the dynamic system. The Kalman filter fails if Q or R equal null matrices, since there is then insufficient data for the development of M(k). The matrix Q can be considered a measure of uncertainty about the system input, rather than a measure of the additive noise affecting the system input. Q can thus be altered strategically to improve the filter's performance. This may be done while the filter is operating.

The Kalman filter gain, M(k), is determined dynamically as the algorithm executes due to time-varying changes in A(k) and C(k). For a linear time-invariant dynamic system, the matrices A, B, C, and D contain constants. Consequently a special case of the Kalman filter algorithm can be developed for application to linear time-invariant systems:

(0) Set $k = 0$
Input A, B, C, D, Q, R, G(0), $x_e(0)$
Set $k = k + 1$

(1) Compute $P(k) = R + CG(k)C^T$ and $P^{-1}(k)$,

(2) Compute $M(k) = AG(k)C^TP - 1(k)$,

(3) Compute $x_e(k + 1)=$
$Ax_e(k) + M(k)[y(k) - Cx_e(k) - Du(k)] + Bu(K)$,

(4) Compute $G(k + 1) = [A - M(k)C]G(k)A^T(k) + Q$,

(5) Set $k = k + 1$
Go to Step (1).

For a linear time-invariant system, the Kalman filter gain M(k) in Step (2) develops in a dynamic manner independent of $x_e(k)$, and the matrices P(k), G(k) and M(k) approach limiting values P^*, G^* and M^* if the underlying dynamic system is stable. These limiting values can be pre-computed off-line and Step (3) can then be used with the matrix M^* as a suboptimal state estimator whose performance is identical to the optimal state estimator except during an initial transient.

To determine the limiting matrix values P^*, G^* and M^* it is sufficient to assume that the time index k is large ($k >> 1$) so that k is approximately equal to $k + 1$, and to solve the following simultaneous matrix equations:

$$P = R + CGC^T \text{ or } P^{-1} = \left(R + CGC^T\right),$$

$$M = AGC^T P^{-1},$$

and

$$G = \left[A - MC\right] GA^T + Q .$$

Figure 6-2 illustrates the performance of a Kalman filter for the following dynamic system:

$$x(k+1) = 0.5\,x(k) + 1.0\,u(k) + w(k)$$

$$y(k) = x(k) + v(k) .$$

The noise processes w(k) and v(k) were, in this example, taken to be white Gaussian noise, both with a variance of 0.1. The input u(k) was a unit step. The presence of the output noise process v(k) makes it impossible to directly measure the state x(k) by means of the output y(k). The presence of the input disturbance w(k) means that the system cannot exactly follow the command input. If the system input were noise free, the output would quickly rise to the final value 2.0.

The solid line in this figure indicates the true, but noisy, nature of the state variable. The dashed line indicates the estimate of the state variable provided by the Kalman filter. Recall that this estimate minimizes the error covariance. This is indicated by the variation of the true state about the estimate over time.

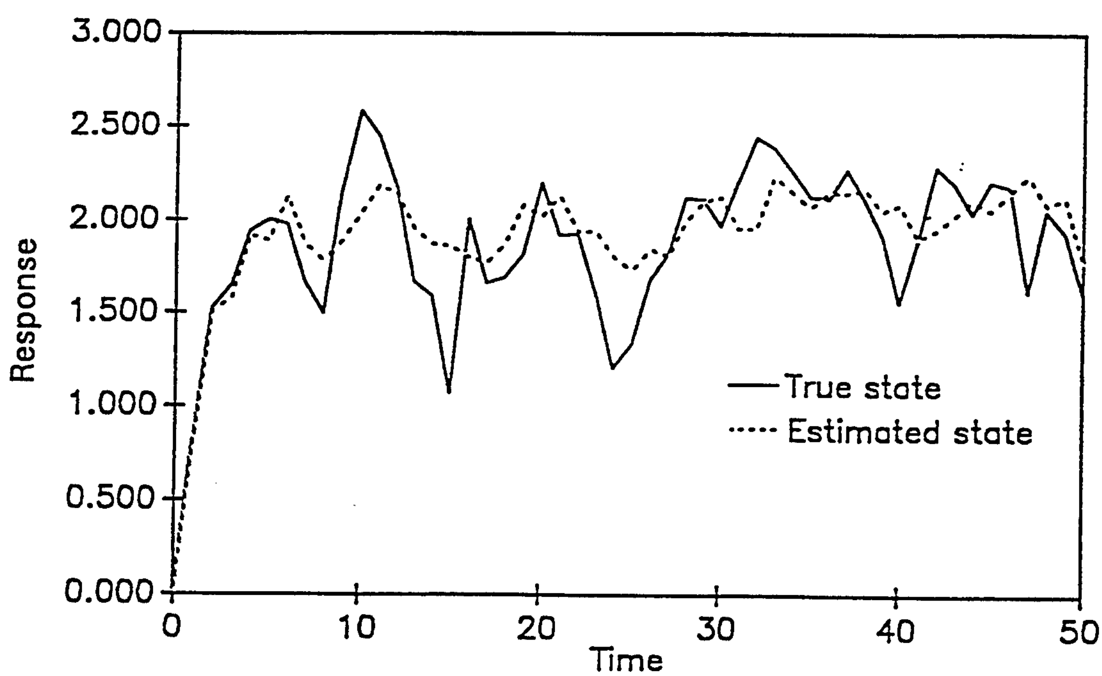


Figure 6-2. Kalman filter performance with an input disturbance w(k).

Figure 6-3 indicates the performance of the filter when the control input u(k) is a sine sequence with an amplitude of 0.5 and a period of 10. Note that the filter again provides a good estimate of the state value.

6.3.2 Split Kalman Filter

The Kalman filter algorithm presented above was designed to estimate the numerical values of the state variables of a discrete-time system. State-variable estimation for a continuous-time dynamic system is handled by selecting a sample time, T, and converting the continuous-time model to a discrete-time model. The sample time T then affects the performance of the discrete-time Kalman filter.

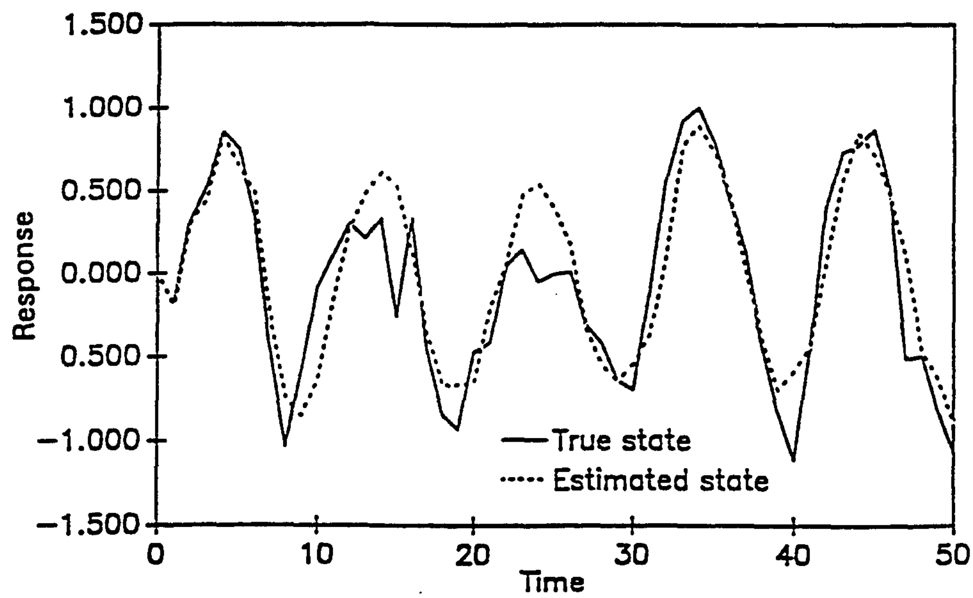


Figure 6-3. Kalman filter performance with a sine sequence control input.

If the sample time is long compared to the operation of the underlying continuous-time system, the estimate $x_e(k+1)$ based on the measured data available at time k can be unacceptably inaccurate. If the most recent data is used to generate the estimated state-variable vector, the performance of the filter can be considerably improved[6.8, 6.9, 6.10].

The split form of the Kalman filter algorithm is:

(0) Set $k = 0$
Input $A(k)$, $B(k)$, $C(k)$, $D(k)$, Q, R, $G_-(0)$, $x_{e-}(0)$
Set $k = k + 1$

(1) Compute $J(k) = G_-(k)C^T(k)[R + C(k)G_-(k)C^T(k)]^{-1}$

(2) Compute $x_{e+}(k) =$
$x_{e-}(k) + J(k)[y(k) - C(k)x_{e-}(k) - D(k)u(k)]$

(3) Compute $x_{e-}(k + 1) = A(k)x_{e+}(k) + B(k)u(k)$

(4) Compute $G_+(k) = [I - J(k)C(k)]G_-(k)$

(5) Compute $G_-(k + 1) = A(k)G_+(k)A^T(k) + Q$

(6) Set $k = k + 1$
Go to Step (1).

In the split form of the Kalman filter algorithm the negative subscript (−) indicates a time just before, and the positive subscript (+) indicates a time just after the sample time k. In this notation $G_-(k)$ is the computed error covariance just before the k^{th} sample time. The gain matrix $J(k)$ is computed and used with the most recent data for $y(k)$ and $u(k)$ to develop the state estimate $x_{e+}(k)$. The computations in Step (2) must be executed quickly following the k^{th} sample time, and the remaining steps must be executed prior to the arrival of the next samples for $u(k)$ and $y(k)$. Two effective computation intervals can be used. The first interval involves the computations of Steps (1), (2) and (4), and the second an interval for computing Steps (3) and (5).

6.3.3 Extended Kalman Filter

Many applications of state-variable estimation involve dynamic systems which are in some way nonlinear, either in terms of the dynamic model or the measurement process. A frequently encountered aerospace model involves a continuous-time nonlinear system:

$$\frac{dx(t)}{dt} = f\big(x(t),\ u(t),\ t\big) + w(t),$$

$$y(t) = g\big(x(t),\ u(t),\ t\big) + v(t),$$

where $x(t)$ is a continuous-time state variable vector, $y(t)$ is a continuous-time measurement vector, and $w(t)$ and $v(t)$ represent noise processes.

These equations can be converted to a set of discrete-time state transition equations by integrating over a single sample time and sampling the input and output:

$$x(k+1) = x(k) + \int_{t=k}^{k+1} T\,f\big(x(t),\ u(t),\ t\big)\,dt + w(k)$$

$$y(k) = g\big(x(k),\ u(k),\ kT\big) + v(k)\ .$$

The resulting discrete-time state transition equations now represent a linear, time-variant dynamic system with an assumed sample time of T seconds, and the split form of the Kalman filter equations can be applied.

The split Kalman filter equations can be placed in one of several groups, those of the measurement update group being evaluated at each sample instant, and those of the time update group being evaluated between sample instants.

The equations of the split Kalman filter measurement update group are:

$$x_{c+}(k) = x_{c-}(k) + J(k)\left[y(k) - g\left(x_{c-}(k),\ u(k),\ kT\right)\right],$$

and

$$G_{+}(k) = \left[I - J(k)\,C(k)\right]G_{-}(k),$$

where

$$C(k) = \frac{\delta g}{\delta x} \text{ evaluated at } x_{c+}(k),\ u(k).$$

The state transition equations are placed in the time update group:

$$x_{c-}(k+1) = x_{c+}(k) + \int_{t=kT}^{k+1} T\,f\left(x(t),\ u(t),\ t\right)dt.$$

For accuracy a numerical integration routine such as a Runge-Kutta algorithm must be used to evaluate this integral. The integration routine will itself require a number of iterations between sample instants. In those cases where the sample time is short compared to the dynamics of the underlying system, a simple rectangular integration process may be used:

$$x_{c-}(k+1) = x_{c+}(k) + T * f\left(x_{c+}(t),\ u(t),\ t\right).$$

An alternative approach is to repeatedly linearize the underlying state transition equations about the present nominal operating point $x + (k)$ and apply the basic Kalman filter equations.

The remaining time update group equations for the extended Kalman filter are:

$$J(k) = G_{-}(k)\,C^{T}(k)\left[R + C(k)\,G_{-}(k)\,C^{T}(k)\right]^{-1},$$

and

$$G_{-}(k+1) = A(k)\,G + (k)A^{T}(k) + Q,$$

where

$$E\left[w(k)\,w^{T}(k+j)\right] = Q\,\delta(j),$$

$$E\left[v(k)\,v^{T}(k+j)\right] = R\,\delta(j),$$

and

$$A(k) = I + \frac{\delta f}{\delta x}, \text{ evaluated at } x_{e+}(k), u(k) .$$

These equations must be evaluated prior to the next sample time, at which the next measured samples arrive.

The steps in the extended Kalman filter algorithm are:

(0) Set $k = 0$
Input Q, R, $G_-(0) = G_+(0) = G(0)$ and $x_{e-}(0) = x_{e+}(0) = x_e(0)$
Compute $C(0) = \delta g/\delta x$, evaluated at $x_{e+}(0)$, $u(0)$

(1) Compute $J(k) = G_-(k)C^T(k)[R + C(k)G_-(k)C^T(k)]^{-1}$

(2) Compute $A(k) = I + \delta f/\delta x$, evaluated at $x_{e+}(k)$, $u(k)$

(3) Compute $G_-(k + 1) = A(k)G_+(k)A^T(k) + Q$

(4) Compute $x_{e+}(k) =$
$x_{e-}(k) + J(k)[y(k) - g(x_{e-}(k), u(k), kT)]$

(5) Recompute $C(k) = \delta g/\delta x$ at the revised $x_{e+}(k)$, $u(k)$

(6) Compute $G_+(k) = [I - J(k)C(k)]G_-(k)$

(7) Compute $x_{e-}(k + 1)=$
$x_{e-}(k)$ + INT[t=kT to (k+1)T] f(x(t), u(t), t) dt

(8) Set $k = k + 1$
Go to Step (1).

An example illustrating the potential effectiveness of the extended Kalman filter applied to a problem of tactical missile guidance and control has been provided by Borrie[6,7]. The problem, involving the pursuit of a target by a surface-to-air missile in two dimensions, is illustrated in Figure 6-4.

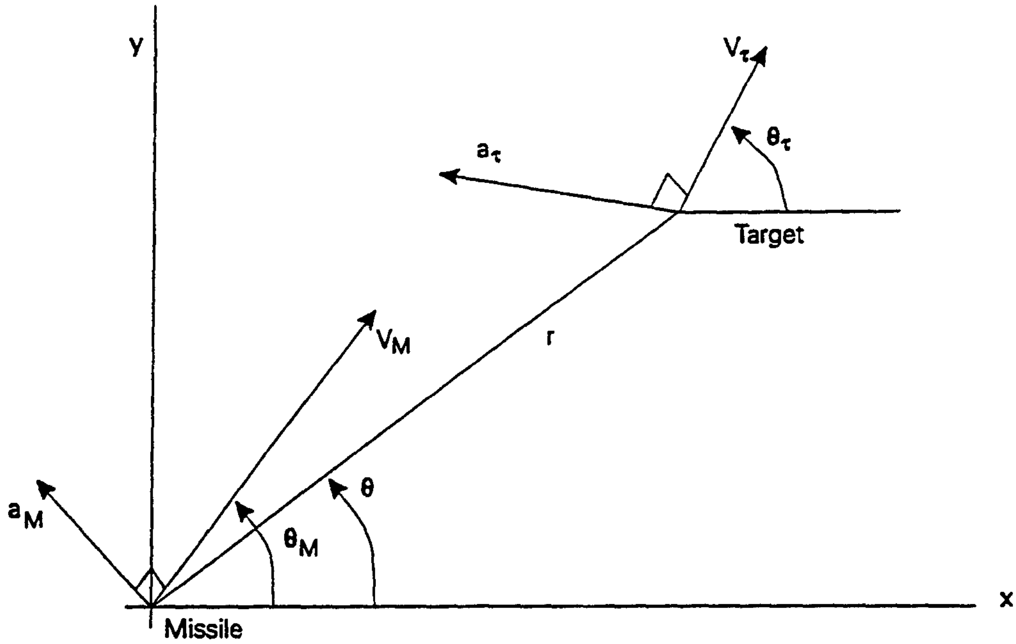


Figure 6-4. Two-dimensional pursuit of a target by a surface-to-air missile.

In this figure,

a_m	=	missile lateral acceleration, m/s^2,
a_t	=	target lateral acceleration, m/s^2,
r	=	missile-to-target range, meters,
Θ_m	=	missile velocity direction, rad,
Θ_t	=	target velocity direction, rad,
σ	=	line-of-sight direction, rad,
V_m	=	missile forward velocity, m/s,
V_t	=	target forward velocity, m/s, and
T	=	sample time, 1 second.

Small variations about the nominal course are assumed, and the missile and target velocities are assumed to be constant. This leads to the following kinematic relationships:

$$\frac{dr}{dt} = V_t\cos(\theta_t-\sigma) - V_m\cos(\theta_m-\sigma)$$

$$r\frac{d\sigma}{dt} = V_t\sin(\sigma_t-\sigma) - V_m\sin(\sigma_m-\sigma) .$$

On the nominal course at constant velocity, $\Theta_m = \Theta_t = \sigma$ and $d^2r/dt^2 = 0$. The rate of change of target lateral acceleration can be modeled as a noise process:

$$\frac{da_t}{dt} = w_t \ .$$

Taking dw_σ/dt, dw_r/dt, w_r and w_t as measures of uncertainty, the following nonlinear state transition equations can be developed:

$$\frac{dx}{dt} = \begin{bmatrix} \frac{-2(x_1\ x_2)}{x_3} - \frac{u}{x_3} + \frac{x_4}{x_3} \\ 0 \\ x_2 \\ 0 \end{bmatrix}$$

$$y = [1\ 0\ 0\ 0]\,x + v_c \ ,$$

where

$$u = a_m \ ,$$

$$x = \left[\frac{d\sigma}{dt}\ \frac{dr}{dt}\ r\ a_t\right]^T$$

and

$$w_c = \left[\frac{dw_\sigma}{dt}\ \frac{dw_r}{dt}\ w_r\ w_t\right]^T \ .$$

When the input signal u is exactly zero, the matrix A(k) required for the extended Kalman filter algorithm is:

$$A(k) = I + \frac{\delta f}{\delta x} \text{ evaluated at } x_{c+}(k),\ u(k) \ ,$$

and at time t = kT this becomes:

$$A(k) = \begin{bmatrix} \frac{-2Tx_{2c+}(k)+1}{x_{3c+}(k)} & \frac{-2Tx_{1c+}(k)}{x_{3c+}(k)} & \frac{2Tx_{1c+}(k)\,x_{2c+}(k) - Tx_{4c+}(k)}{\left(x_{3c+}(k)\right)^2} & \frac{T}{x_{3c+}(k)} \\ 0 & 1 & 0 & 0 \\ 0 & T & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \ .$$

Also, C(k) = [1 0 0 0], and the noise process is developed in discrete-time form by means of:

$$Q_d(k) = \int_{t=0}^{t=T} \exp(At)\, Q_c \left(\exp(At)\right)^T dt \ ,$$

$$Q_d(k) \approx T\, Q_c(t) \; ,$$

where

$$Q_c(t) = E\left[w_c(t)\, w_c^T(t)\right] .$$

Evaluating these expressions,

$$Q_d = Q = \begin{bmatrix} 360 & 0 & 0 & 0 \\ r & & & \\ 0 & 20 & 0 & 0 \\ 0 & 20 & 0 & 0 \\ 0 & 0 & 20 & 0 \\ 0 & 0 & 0 & 4800 \end{bmatrix} .$$

Also,

$$R_d(k) = R_c(kT) \; ,$$

where

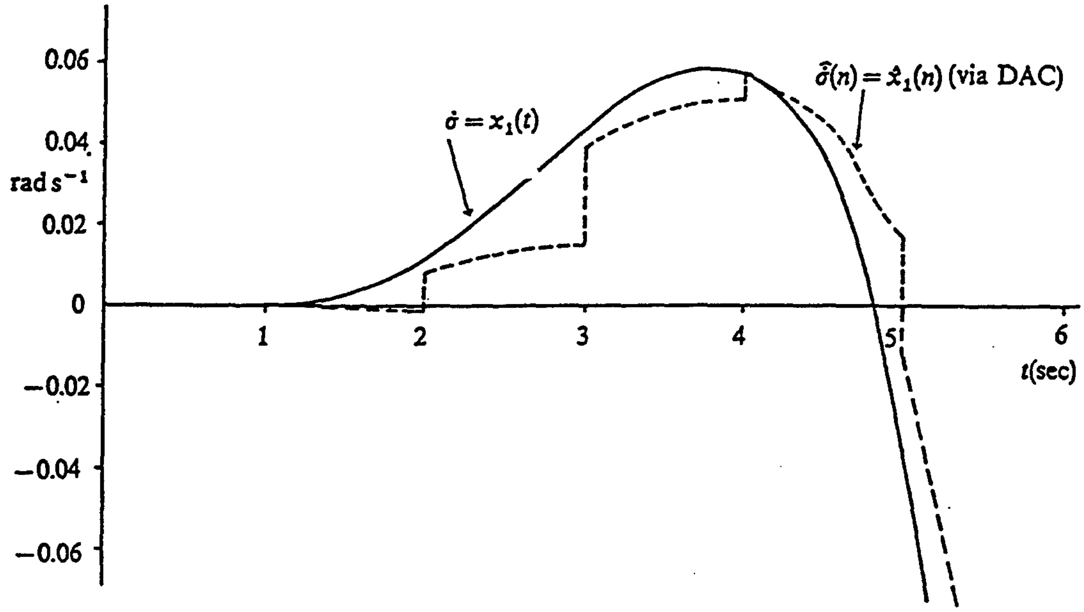
$$R_c(t) = E\left[v_c(t)\, v_c^T(t)\right] .$$

In this example,

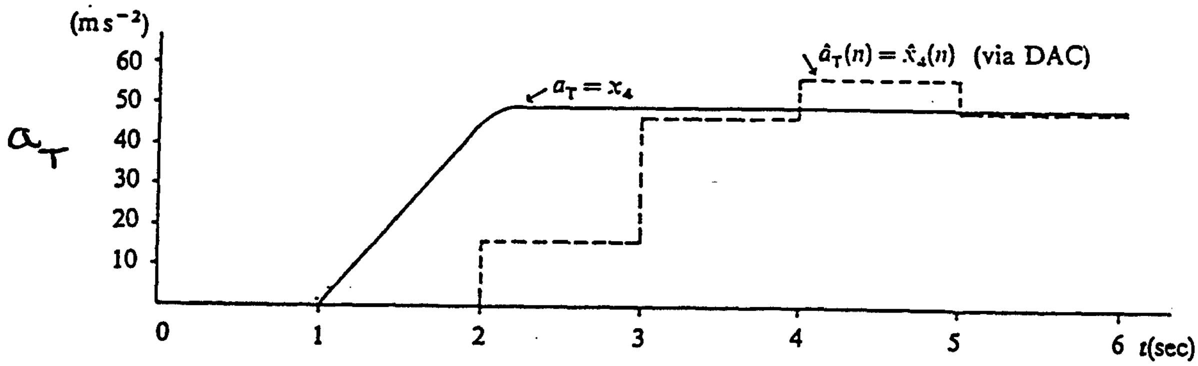
$$R_d = R = \frac{360}{r^2} .$$

Substituting these quantities into the equations for the extended Kalman filter algorithm produced the estimates shown in Figure 6-5. This figure illustrates the continuous-time states $x_1(t)$, the rate of change of the line-of-sight angle σ, and $x_4(t)$, the target acceleration, and the estimated values for these two state variables, for a 50 m/s^2 change in lateral target acceleration. The continuous curves indicate the continuous-time state trajectory, and the dashed curves indicate the estimated state trajectories produced by the extended Kalman filter. Note how the estimate improves after each sample instant, at which times more reliable data became available.

Borrie noted that a natural use for an extended Kalman filter of this form was in a control system for a tactical guided missile intended to yield an improvement over classical proportional navigation.



(a) rate of change of line of sight angle, $\dot{\sigma}$



(b) target acceleration

Figure 6-5. Extended Kalman filter behavior for the example in Figure 6-4.

6.3.4 Noise and Process Modeling

The Kalman filter assumes that the state and measurement noise processes are modeled by white Gaussian noise. In many applications these assumed random inputs may be time-correlated and can be modeled by first- or second-order Gauss-Markov processes. The dynamic system may also contain parameters whose actual values are imprecisely known. One approach to handling these apparent complications is to augment the original state variable vector with additional state variables representing the random parameters and the noise processes, at the expense of a substantial increase in computational burden.

The dynamic model parameters, noise statistics, and initial state estimates may differ substantially from those of the actual system which generates the observations in a particular application of the Kalman filter. It may occasionally be desirable to use a dynamic system model of lower dimension than the original system, thus reducing computational requirements. This reduced-order model may or may not degrade the performance of the control system. Algorithms have been developed and are available to analyze the mean square error performance when a dynamic model other than that of the original system is used. The reduced-order models required for this procedure are developed using the methods of dynamic system identification.

6.3.5 Kalman Filter Divergence

Round-off and truncation errors which occur automatically during computations and system modeling errors made during the analysis and design of a Kalman filter are known to reduce the performance of a Kalman filter below that predicted by analysis. After many computation cycles, the effects of these errors may be intolerable. Generally, the covariance matrix entries are too small as a result of these errors, and the filter gains are then automatically reduced to the point where subsequent observations are ignored. Two methods which may be tried to avoid the problem of filter divergence are the use of a numerically greater process noise covariance, which compensates for modeling errors and improves the filter's stability, and the use of exponential data weighting, which prevents obsolete data from saturating the filter.

Computer round-off errors in the covariance propagation equations can result in filter divergence. The Kalman filter equations generally require the use of double-precision computations. Several algorithms based on matrix square root propagation have been developed and these algorithms yield twice the effective precision as the conventional Kalman filter.

6.3.6 Kalman Filter Design Considerations

The process of designing a Kalman filter is an iterative one, involving considerable physical insight and tradeoffs between the computational burden, the filter requirements, and the expected performance. A significant analytical effort is required to reach a final, well-tuned design. Maybeck[6.10] suggests the following Kalman filter design procedure:

(1) Develop a dynamic model to represent the system of interest and validate the operation of this model with experimental data. A physically-based model that is highly complex may not be required, but a mathematical model which represents the system dynamics to a sufficient level of accuracy is a necessity.

(2) Design the Kalman filter by applying and implementing the filter equations directly to establish a performance benchmark, and compare the resulting estimation errors against the filter specifications. A study of the computational constraints may reveal ways in which a different choice of coordinate system and a different definition of the system state variables can reduce the Kalman filter's complexity.

(3) Propose a set of reduced-order Kalman filters by combining and deleting states and removing weak cross-couplings. Evaluate possible approximations to the computed optimal filter gain, perhaps by the steady-state gain matrix.

(4) Conduct Monte Carlo simulations of the proposed filters' performance and conduct a covariance matrix sensitivity study.

(5) Select a final design based on the required performance and computational requirements.

(6) Perform checkout, final tuning, and an operational test of the filter.

In any implementation of the Kalman filter the largest computational burden will arise as a result of the computation of the transition matrix and the propagation of the error covariance matrix. These quantities are not required to have the same level of accuracy as the estimated state vector, and so they may be updated at a slower rate than the state vector.

The lack of statistical information about the process and measurement noise is a common difficulty in most applications of the Kalman filter. Often, little data regarding the standard deviations and correlation time constants will be available. The observation process seldom behaves in the same manner as the idealized process. For example, the observations may be relatively noise-free for a long period of time, then become highly contaminated. Linearization effects, modeling errors, and the use of reduced-order filters all influence the overall operation of the control system. The error covariance matrix computed from the Kalman filter Ricatti equation may not accurately represent the true covariance matrix at any one point in time. It is generally necessary to conduct extensive simulations to study and establish confidence in the Kalman filter's performance in any particular application.

6.3.7 Application to Target Tracking

A Kalman filter can be used in a tracking application to estimate the position, velocity, and acceleration of a maneuvering target from a set of noisy observations that is applicable to all weapon systems. The target may be a surface craft or vehicle, an aircraft, or a missile. A sensor system consisting of a radar, sonar, or optical means for measuring the target's range, azimuth, and elevation at a high data rate is assumed. The measurements provided may also include range rate data.

To design a tracking filter one must rely on the basic laws of motion and a stochastic acceleration model for the target. Usually it is assumed that the target moves at a constant velocity if there is no maneuver occurring or if there is no atmospheric turbulence. Disturbances and evasive maneuvers are considered to be perturbations on a constant velocity trajectory.

The acceleration of a target is known to be correlated in time. For example, an aerial target in a smooth, gradual turn will exhibit a highly correlated acceleration. Evasive maneuvers result from target accelerations which are less correlated in time. A correlation function for the target acceleration is:

$$\gamma(\tau) = E\left[a(t)\ a(t+\tau)\right] = \sigma_m^2 \exp\left(-\alpha \ \mathrm{abs}\,(\tau)\right)$$

where

E[ ] denotes the expected value operation,

a(t) is the target acceleration at time t,

σ^2 is the target acceleration variance, and

$1^m/\alpha$ = maneuver time constant.

The target motion dynamic model in one dimension is:

$$\frac{d}{dt}\begin{bmatrix} p \\ v \\ a \end{bmatrix} = \begin{bmatrix} 0 & 2 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & -\alpha \end{bmatrix} \begin{bmatrix} p \\ v \\ a \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} w(t)$$

where

p = p(t) = target position,

v = v(t) = target velocity,

a	=	$a(t)$ = target acceleration, and
$w(t)$	=	a white Gaussian noise process with zero mean and finite variance.

The uncertainty in the target acceleration is modeled by a first-order Gauss Markov process.

This filtering process can also be performed in the polar coordinates of range, azimuth, and elevation with their derivatives as state variables. In polar coordinates the measurement model would be linear and uncoupled, since range azimuth and elevation are measured independently. In either case, for a target moving in a straight line at constant speed, the multi-dimensional dynamic model is nonlinear, and the use of a linearized motion model leads to large errors in the predicted target position. In rectangular coordinates, a constant speed target can be modeled by a linear motion model, but the measurement equations are then nonlinear. The choice of coordinate system, motion model, and measurement model is up to the system designer, and depends on the sensor suite.

Full implementation of a Kalman filter for a process with n state variables requires an ability to solve an n by n matrix Ricatti equation on-line. The use of a microprocessor with a restricted word length can be expected to introduce numerical errors due to truncation. A common design approach intended to reduce the computational burden relies on neglecting selected cross-coupling terms and pre-computing those filter gains which do not significantly change. Generally, extensive simulations are required to tune the resulting suboptimal Kalman filters and obtain an appropriate level of performance.

Target state estimation is a normal part of tactical missile guidance and control system design. The objective is to estimate in the best possible way the position, velocity, and acceleration of the target based on observations provided by the missile's seeker. These estimates are then used to predict the future trajectory of the target and improve the intercept.

A related successful application which illustrates the process of state estimation is the estimation and control of spacecraft attitude. The angular velocity of the spacecraft is obtained from a combination of on-board gyroscopic sensors and a stabilized inertial platform. Sun sensors and star trackers are used to measure the spacecraft's attitude. The kinematic equations which define this system are processed to obtain the multivariable attitude state, and this state is augmented by additional state components which define the biases in each gyro. In this implementation, the gyro data is not treated as an observation and the gyro noise appears as state noise rather than as observation noise.

To relate the reference axes of the spacecraft to the inertial axes, a set of three rotation angles called the Euler angles are normally used. The Euler angles represent the relative pitch, yaw, and roll angles of the spacecraft, and have easily interpreted physical meanings. However, the dynamic model representing these angles and their rates is nonlinear and contains several complicated trigonometric terms. Additionally, the Euler angles become undefined for some rotations, i.e., gimbal lock conditions. For these reasons, an alternate representation is preferred.

One possible representation which avoids the direct use of Euler angles is the use of direction cosines to measure the spacecraft attitude. This model, however, results in a non-orthogonal attitude matrix due to round-off errors, quantization errors, and truncation errors encountered during the model's evaluation. Since the computational burden required to compensate for these predictable errors and to implement the redundant nine parameter direction cosine model is relatively high, this approach is not widely used.

A better choice for representing the spacecraft's attitude is a state variable format based on a set of four parameters comprising a quaternion. This quaternion can be analytically derived in a straightforward manner, and propagated in time by a set of four first-order linear differential equations. The direction cosines, and the resulting Euler angles, can be computed as quadratic forms of the quaternion. The elements of the quaternion do not have a direct physical interpretation, but their computational advantages outweigh this aspect of their use. One difficulty with the use of the quaternion representation and Kalman filtering to estimate its components involves the constraint that the quaternion vector has a unit norm. Efforts to propagate the state vector and covariance matrix and re-normalize the quaternion vector have been described by Lefferts et.al[6.11].

6.4 Summary

Modern control theory makes use of estimation procedures that are generally stochastic. The estimation problem consists of five components: a set of unknown variables or parameters, a family of statistical probability functions, a scalar or vector random variable, a set of estimator functions, and a loss function. Bayesian and non-Bayesian approaches are used to solve parameter estimation problems. Both approaches may be applied to linear or non-linear estimations. A special case of a linear system that has application to guidance and control is the so-called Kalman filter. Most of this chapter was devoted to discussion of the Kalman filter.

Three components make up the Kalman filtering process: knowledge of the system model, a set of a priori estimates, and some measure of white noise in the system. The Kalman filter algorithm was described. Special applications to discrete-time and non-linear systems were also discussed. A specific application to the two-dimensional pursuit of a target by a surface-to-air missile was included. The chapter was concluded with some general Kalman filter design considerations and their use in target tracking.

REFERENCES

6.1 Sage, A. P., and Melsa, J. L., Estimation Theory with Applications to Communications and Control, McGraw-Hill Book Company, New York, 1971.

6.2 Kalman, R. E., and Bucy, R. S., New Results in Linear Filtering and Prediction Theory, Journal of Basic Engineering, Vol. 83, pp. 95-108, 1961.

6.3 Kallianpur, G., Stochastic Filtering Theory, Springer-Verlag, New York, 1980.

6.4 Wong, E. and Hajek, B., Stochastic Processes in Engineering Systems, Springer-Verlag, New York, 1985.

6.5 Sorenson, H. W. (Ed.), Special Issue on Applications of Kalman Filtering, IEEE Transactions on Automatic Control, Vol. TAC-28, pp. 253-434, 1983.

6.6 Schmidt, S. F., The Kalman Filter: Its Recognition and Development for Aerospace Applications, Journal of Guidance, Control, and Dynamics, Vol. 4, pp. 4-7, 1981.

6.7 Borrie, J. A., Modern Control Systems, Prentice Hall International, Englewood Cliffs, New Jersey, 1986.

6.8 Leondes, C. T., Theory and Applications of Kalman Filtering, AGARDograph No. 139, AGARD, 1970.

6.9 Leondes, C. T., Advances in the Techniques and Technology of the Applications of Nonlinear Filters and Kalman Filters, AGARDograph No. 256, AGARD, 1982.1982.

6.10 Maybeck, P. S., Stochastic Models, Estimation and Control, Vols. I, II, III, Academic Press, New York, 1982.

6.11 Lefferts, E. J., Markley, F. L., and Shuster, M. D., Kalman Filtering for Spacecraft Attitude Estimation, Journal of Guidance, Control, and Dynamics, Vol. 5, pp. 417-429, 1982.

This page is intentionally blank.

CHAPTER 7
ADAPTIVE CONTROL

7.1 Introduction

If, for a linear, time-invariant, single-input, single-output dynamic system, the mathematical model structure, the numerical values of the model's parameters, and the characteristics of the noise or disturbance affecting the system are known, classical control system design methods using frequency or time domain techniques can be used to derive a fixed-structure, constant-parameter control system which will regulate the system about some desired state or permit the system to track an external input signal with reasonable accuracy. When the parameters of the dynamic system vary over wide ranges, the performance of a fixed control system design will generally be unsatisfactory and some form of adaptive compensation will be required. A control system which actively measures and collects information about the dynamic system's performance and the noise incurred and uses that information on-line to alter the structure or parameters of the control system itself is called an adaptive control system[7.1].

Adaptive control involves sensing one or more system variables and using that sensed data to vary the structure of a feedback control system. The objective of an adaptive control strategy is to improve the system's performance compared to that obtained using a fixed control structure. There are several related and overlapping techniques which comprise the technology of adaptive control, including gain scheduling, model reference adaptive control, the self-tuning regulator, and control system designs based on optimal control theory.

A modern control system designer has three options available when designing a new control system. A fixed controller might be selected, in which the structure, gains, time constants, and other parameters are selected and built or programmed into the device during its construction. A fixed controller is generally acceptable if the system dynamics do not change appreciably over time. For example, two simple gain blocks are all that is required to close the loop and stabilize the ideal double integrator shown in Figure 7-1. By varying the feedback gains and thus selecting the location of the closed-loop poles, the designer can obtain a wide range of performance characteristics and reduce sensitivity to variations in the gains themselves.

If changes in the system dynamics can be predicted on the basis of measurable data about the environment, then a gain scheduling approach can provide an improvement in control system

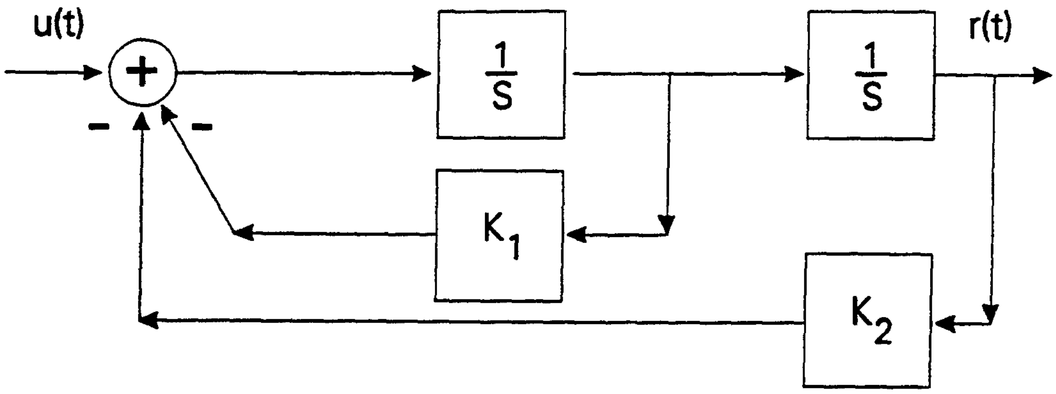


Figure 7-1. Stabilized double integrator.

performance compared to a fixed controller. This method has been widely applied to develop and implement autopilots for aerodynamic missiles. The missile dynamics are predictable and are known to change as a function of dynamic pressure, a quantity which involves the missile velocity and altitude. Since dynamic pressure can be measured directly or inferred from on-board sensors, the autopilot gain can be made a function of dynamic pressure. The gain values can be determined by repeating the control system analysis for several operating altitudes and velocities, and a table of gain versus dynamic pressure constructed.

Fully adaptive control, the modern designer's third option, can be considered an implementation of a continuous process of control system design. The adaptive control approach is highly applicable if the system dynamics are known to vary over a significant range, or if measurable but uncertain external effects must be accounted for. Adaptive control can also be implemented in response to a diagnostic procedure. This approach permits a control system to reconfigure itself and may, for example, allow a damaged tactical missile to continue on its mission.

When the parameters of a dynamic system are unknown or their values fluctuate widely due to manufacturing tolerances, external influences, or other uncertainties, the dynamic system cannot be modeled as a time-invariant linear system, and most methods of classical control system synthesis cannot be applied. If a control system is designed on a classical basis, without the capability to adapt to unpredictable changes, the performance obtained when unpredictable changes occur is likely to be degraded below design specifications.

Feedback is used in conventional single-input, single-output control systems to reject the effect of disturbances on the output and to bring the system outputs back to their desired values. The feedback concept can also be applied to the problem of maintaining the performance of a closed-loop control system faced with parameter changes or other unpredictable disturbances. A performance measure for the closed-loop system must first be defined. The measured performance is compared to the desired performance and the difference serves as the input to an adaptation mechanism. The adaptation mechanism produces an output which is used to modify the parameters of the control system, or the control input, which in turn modifies and corrects the performance of the composite system. Figure 7-2 illustrates the basic concept of an adaptive control system.

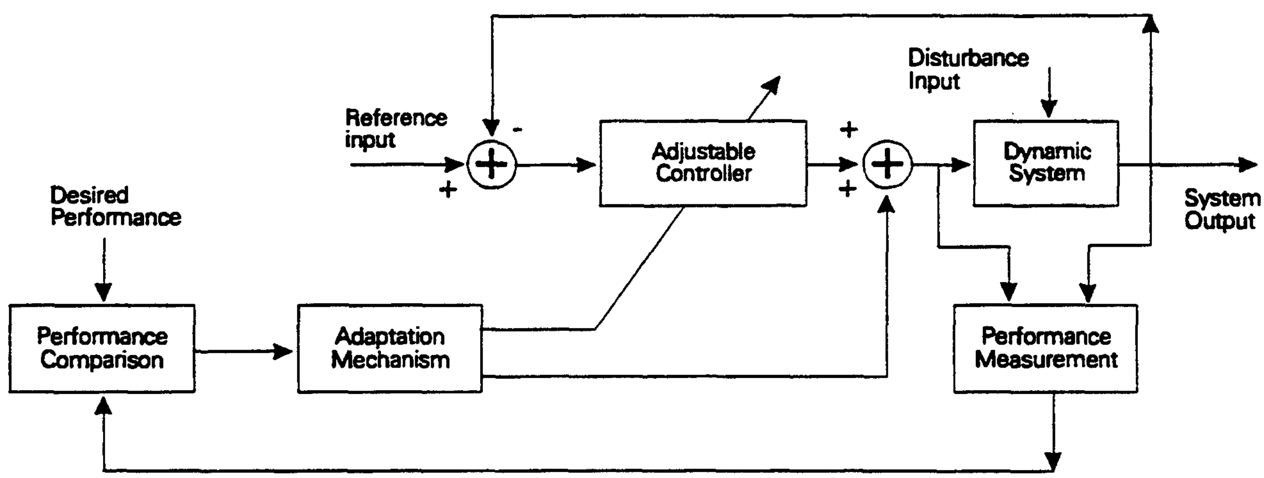


Figure 7-2. Basic adaptive control system configuration.

One interpretation of an adaptive control system is that of a feedback control system where the ultimate controlled variable, or output, is the performance measure. Figure 7-2 indicates that the adaptive control loop appears as an additional feedback loop which is implemented and allowed to function whenever the basic feedback control system requires its performance to be monitored and improved.

7.2 Applications of Adaptive Control

The primary application of adaptive control theory to tactical guided weapons relates to the design of closed-loop control systems and components such as seekers, guidance computers, autopilots, and actuators. In a classical design approach, methods such as gain scheduling may be used to vary autopilot gains in response to Mach number, altitude, and other factors. Adaptive control theory allows a designer to extend this approach by providing control systems which not only sense and compensate for changes in the environment, but also sense and compensate for changes in the weapon system itself. For example, an adaptive control system may be able to reconfigure itself to compensate for a partially disabled actuator, thus allowing the weapon to successfully complete its mission.

The control components of interest in a tactical weapon system are usually servo systems, and the input signal or reference input is normally generated as a result of seeker computations. In a classical design approach, a nominal control system design is specified, based on a set of design computations which span the expected operating range of the system. Once designed in this manner, the various gains, time constants, limits, and other parameters of the control system cannot be changed to reflect unanticipated external conditions or hardware reliability. This technological situation contrasts with the situation in the process control industries, where the control system is continuously monitored by an operator who has the freedom to readjust, tune, and possibly reconfigure the control system to optimize the performance processes. The closed-loop control systems of today's tactical weapons must operate with virtually no operator intervention, and this provides an opportunity for the application of adaptive control technologies.

The availability of additional computational power on-board a tactical missile or other weapon system permits the inclusion of self-test and diagnostic capabilities not found in systems implemented with traditional analog control systems. The same sensors which are required to implement an adaptive control methodology may also be used in a self-test mode of operation. In this mode, a weapon could continually monitor its own readiness, and automatically report any change in its status, prior to its attempted use.

7.3 Overview of Adaptive Control Methods

A method proposed early on for adaptive control uses auxiliary variables which are correlated in some way with parameter changes in the dynamic system. This heuristic approach is called gain scheduling by Stein[7.2]. The basic idea is to compensate for system parameter variations by changing the controller gains or other parameters. The use of an auxiliary variable allows the design of a look-up table or other device to provide the appropriate gain. Gain scheduling is now commonly used to vary missile autopilot gains as a function of altitude, Mach number, dynamic pressure, or some other easily measured variable. Gain scheduling can be thought of as a form of classical feedforward compensation. The general structure of a gain scheduling control system is shown in Figure 7-3.

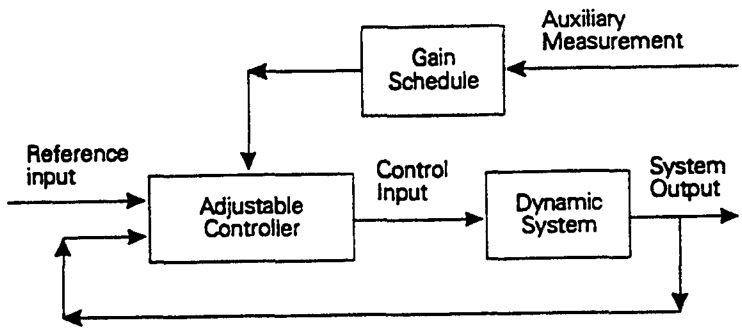


Figure 7-3. System with gain scheduling.

Adaptive control methods can be used to automatically adjust the parameters of a controller used for a high-performance tactical missile system in which the parameters of the controlled dynamic system are uncertain, unknown, or expected to vary widely during normal operation. Model reference adaptive control is one adaptive control technique which is relatively easy to implement, offers the potential for improved system performance in a wide variety of applications, and for which design procedures now exist[7.3,7.4]. A model reference adaptive control system is shown in Figure 7-4.

In a model reference adaptive control system, the system performance specifications are given in terms of a reference model, a mathematical description of the ideal behavior of the dynamic system. The model is implemented as part of the overall control system and indicates how the dynamic system should ideally respond to a command input. The model reference control system consists of two separate loops, an inner loop, a classical feedback loop consisting of the dynamic system being controlled and an adjustable regulator or compensator, and an outer loop which adjusts the parameters, or gains, of the regulator or compensator in response to sensed changes in the dynamic system parameters.

Figure 7-5 illustrates a self-tuning regulator. The self-tuning regulator consists of two control loops. The inner control loop consists of the dynamic system and an adjustable regulator. The outer loop consists of a parameter estimator which continually updates a mathematical model of the dynamic system and a control system design mechanism which yields an updated design for the regulator. The self-tuning regulator automates the two concurrent processes of dynamic system identification and control system design.

Any of the classical design methods can be used to implement the design process for a self-tuning regulator and many methods for parameter estimation have been developed. Methods for classical control system design were briefly discussed in Chapter 2 and techniques for dynamic system

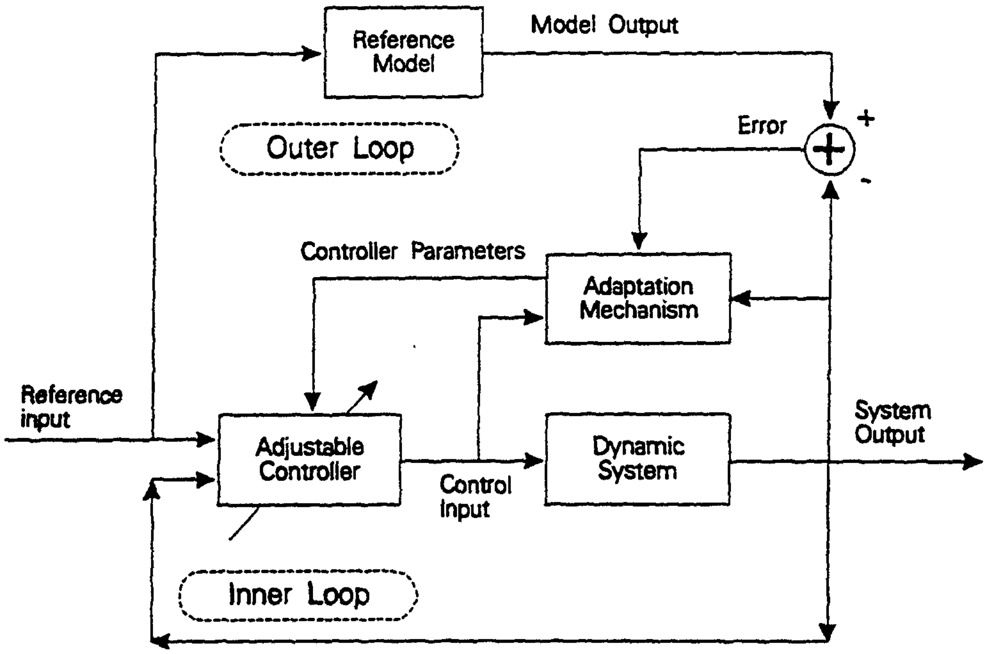


Figure 7-4. Model reference adaptive control system.

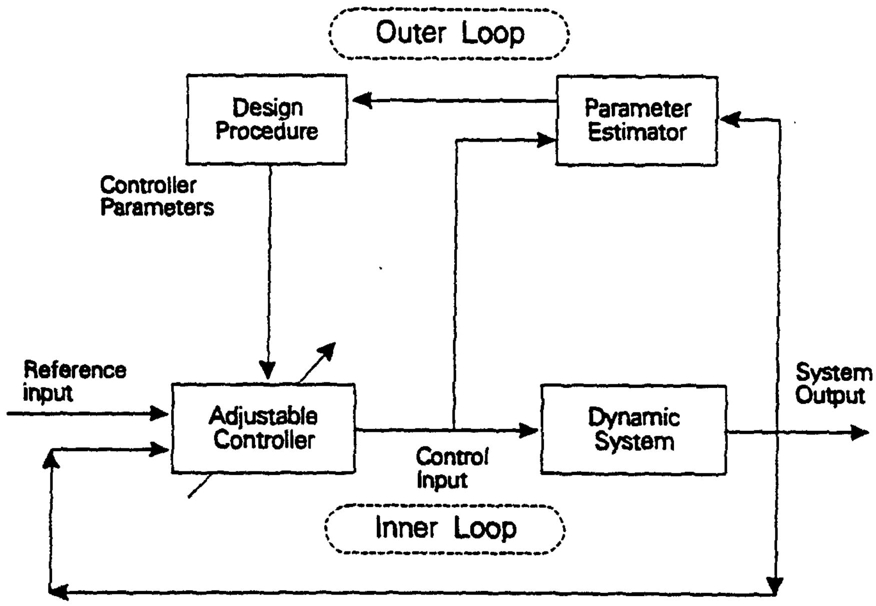


Figure 7-5. Self-tuning regulator adaptive system.

identification in Chapter 5. For example, the design calculation may be based on classical gain and phase margins, pole placement, or linear quadratic optimal control theory. The identifier or estimator may be based on a least-squares method, an extended Kalman filter, the maximum likelihood method, or any other technique for evaluating the required set of model parameters or system states.

The model reference control system shown in Figure 7-4 is referred to as a direct adaptive control method because the regulator's parameters are updated directly as a function of any changes in the dynamic system's parameters. The self-tuning control system shown in Figure 7-5 is referred to as an indirect adaptive control method because the regulator's parameters are updated indirectly, based on recursive design computations. An indirect method may be converted to a direct method by mathematically implementing the design computations in terms of the estimated system parameters.

Model reference adaptive systems also have other uses, including adaptive prediction and

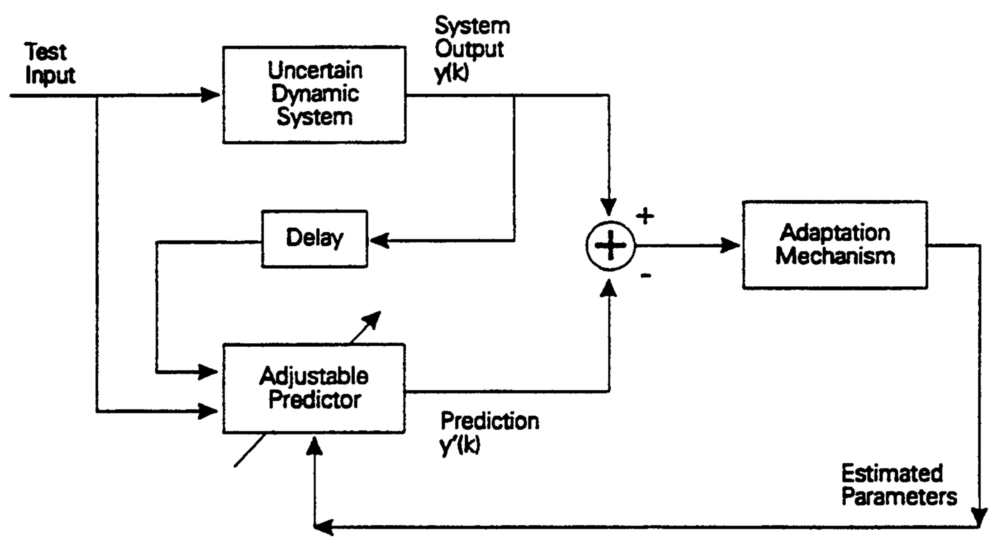


Figure 7-6. Adaptive prediction and parameter estimation.

parameter estimation. This concept is illustrated in Figure 7-6.

The dynamic system of Figure 7-4 with its uncertain parameters represents the reference model. The adjustable predictor is the adaptive system in Figure 7-6. The purpose of the adaptation mechanism is to force the parameters of the adjustable predictor to those values which asymptotically drive the prediction error to zero in a deterministic environment. At the end of this process, the resulting parameter values form a best estimate model of the unknown dynamic system. A model of the uncertain dynamic system will be obtained in which the input-output relationship compared to that of the dynamic system is as accurate as possible for the specified test input sequence. The adaptation mechanism shown in Figure 7-6 can be used to implement recursive or on-line system identification.

7.4 Model Reference Adaptive Control Systems

There has been continuing interest in the application of adaptive control technology since the early 1960s. Many different and ingenious approaches to the development of adaptive control systems have emerged. Most designs have been effective, subject to technical assumptions regarding the specific dynamic system and its operating environment. There is now a well developed theory for a special class of adaptive control systems, model reference adaptive control systems[7-5].

The model reference adaptive control method attempts to determine in real time the parameters of an adjustable controller such that the response of the resulting closed-loop system consisting of the adjustable controller and the underlying dynamic system is identical to the response of a reference model. The reference model is a mathematical model whose structure is generally less complicated than that of the underlying dynamic system. The structure of a model reference control system was shown in Figure 7-4.

The theory of model reference adaptive control systems and the results of its application have appeared over the past decade. The design approach for model reference adaptive controllers is based on the stability theories of Lyapunov and Popov. The approach involves the design and implementation of intentionally nonlinear controllers which ensure the global stability of the overall system comprised of the dynamic system and the adaptive controller. Model reference adaptive control systems have successfully been applied to the design of servo systems characterized by low noise levels and a knowledge of the dynamic system sufficient to guarantee the absence of right-half plane zeros.

7.4.1 Model Reference Adaptive Control Design Parameters

In a model reference adaptive control system, the desired system performance is usually specified in terms of pole-zero locations or a transfer function matrix. The desired pole-zero configuration permits the design of a mathematical reference model which generates the desired behavior of each of the system's outputs. The errors between the outputs of the reference model and the outputs of the dynamic system are used as inputs to an adaptation mechanism which performs, on-line, the computations necessary to adjust the parameters of the controller and bring the performance of the actual dynamic system into close agreement with the specified reference model performance.

The control system shown in Figure 7-4 is an example of an explicit model reference adaptive control system. In an explicit adaptive controller, the reference model is part of the adaptive control loop. The difference between the output of the dynamic system and the output of the reference model is a measure of the difference between the real and the desired dynamic system performance. This difference, called the model error, is used by the adaptation mechanism along with other information to automatically adjust the parameters of the controller and asymptotically drive the model error to zero in a deterministic environment having no other external random disturbances.

A parameter adaptation algorithm is used to modify the controller parameters in response to the model error. This algorithm forms the basis of the adaptation mechanism which attempts to continually reduce the error between the computed reference model output and the actual dynamic system output. The key problem in the design of a model reference adaptive control system is to find a means for adjusting the controller parameters. In general, this cannot be implemented by classical linear error feedback. One rule which successfully worked in early applications of model reference control systems was:

$$\frac{dI}{dt} = -\alpha \cdot e \, \nabla_1 [e] ,$$

where

I	=	a vector of controller parameters,
α	=	the adaptation rate,
e	=	the model error, and
$\nabla_1 [e]$	=	the partial derivative of the model error with respect to the controller parameters.

In a model reference adaptive control system, the reference model must be carefully selected so that its performance can be duplicated by the underlying dynamic system when driven by the controller-produced inputs. The dynamic system may not have any zeros in the right-hand plane because the controller effectively cancels out plant zeros and replaces them by the zeros of the reference model. This approach can lead to instability if cancellations of right-hand plane poles and zeros occur.

7.4.2 Model Reference Adaptive Control System Design Example

The general principle involved in the design of a model reference adaptive control system is the use of a reference model selected by the designer and characterizing the desired system performance. A composite system consisting of the underlying dynamic system and a controller with adjustable parameters is constructed. The design goal is to continuously adjust the controller parameters so that the composite system performs as desired. The reference model is specified by the designer such that the output of the reference model defines the performance desired of the dynamic system. The problem is to develop a rule, or algorithm, which adapts the parameters of the adjustable controller so that the output of the underlying dynamic system tracks the output of the reference model.

When formulated in this manner, the principles behind a model reference control system can also be applied to other problems in the system identification and control area[7.6]. Parks[7.7] pioneered the use of stability theory to design the adaptation mechanism, and was the first to exploit the properties of positive real transfer functions in the development of model reference adaptive control systems. The general process of developing a model reference adaptive controller is illustrated in the following example.

A simple first-order dynamic system has a known time constant, T, but an unknown gain, K_p. The uncertainty in K_p may arise due to external influences, component ageing, a time-varying parameter, manufacturing tolerances, or other factors not under the control of the designer. The desired relation between the control input u(t) and the system output y(t) is defined by a reference model with output $y_m(t)$. The reference model has a specified time constant, also T, and a specified gain, K_m.

The dynamic system is represented mathematically by the following differential equation:

$$\frac{dy(t)}{dt} = -a\,y(t) + b\,u(t)$$

and the reference model is similarly represented by:

$$\frac{dy_m(t)}{dt} = -a_m\,y_m(t) + b\,u_c(t) \ ,$$

where

$$a = \frac{1}{T},\ b = \frac{K_p}{T},\ a_m = \frac{1}{T} \text{ and } b_m = \frac{K_m}{T} \ .$$

A simple algebraic substitution shows that perfect model following can be obtained using the adaptive controller which provides the modified input:

$$u(t) = t_0\,u_c(t) - s_0\,y(t) \ ,$$

where the parameters are:

$$t_0 = \frac{b_m}{b},\ s_0 = \frac{(a_m - a)}{b} \ .$$

The system output $y(t)$ is presumed to be measurable, as is the command input $u_c(t)$ which feeds the reference model.

The objective is to adjust the parameters t_0 and s_0 so that the model error defined by $e(t) = y(t) - y_m(t)$ tends to be zero. A solution can be constructed by defining a Lyapunov function, one example of which is:

$$V = \frac{1}{2}\left[e^2(t) + \left[\frac{1}{bc}\right](bs_0+a-a_m)^2 + \left[\frac{1}{bc}\right](bt_0-b_m)^2\right]$$

where $c > 0$. This function is zero when the output error is zero and when the controller parameters have been exactly adapted to the uncertain dynamic system. If the parameters are adjusted according to:

$$\frac{dt_0(t)}{dt} = -c\,u_c(t)\,e(t) \ ,$$

$$\frac{ds_0(t)}{dt} = +c\,y(t)\,e(t) \ ,$$

the time derivative of the Lyapunov function is:

$$\frac{dV(t)}{dt} = -a_m\,e^2(t) \ ,$$

and is negative-definite, indicating that the system is stable, and that the error $e(t)$ tends to zero as time increases. The value of the parameter c can be selected to control the adaptation rate. The results for a time constant T equal to one second, a reference model gain K equal to 1.0 and a time-varying dynamic system gain K_p modeled by:

$$K_p = 1.0 + 0.2\sin\left[\frac{2\pi t}{25}\right] \ ,$$

a value of 7.0 for the parameter c are shown in Figures 7-7, 7-8, and 7-9. Figure 7-7 shows the transient step responses of the time invariant reference model and the time varying dynamic system. Note that the controlled dynamic system closely follows the reference model. The transient response of the uncontrolled dynamic system is also shown for comparison. Note the continuing divergence of the uncontrolled response as compared to the reference model output. The values of the time varying parameter b are shown in Figure 7-8 and the time varying controller gains are shown in Figure 7-9.

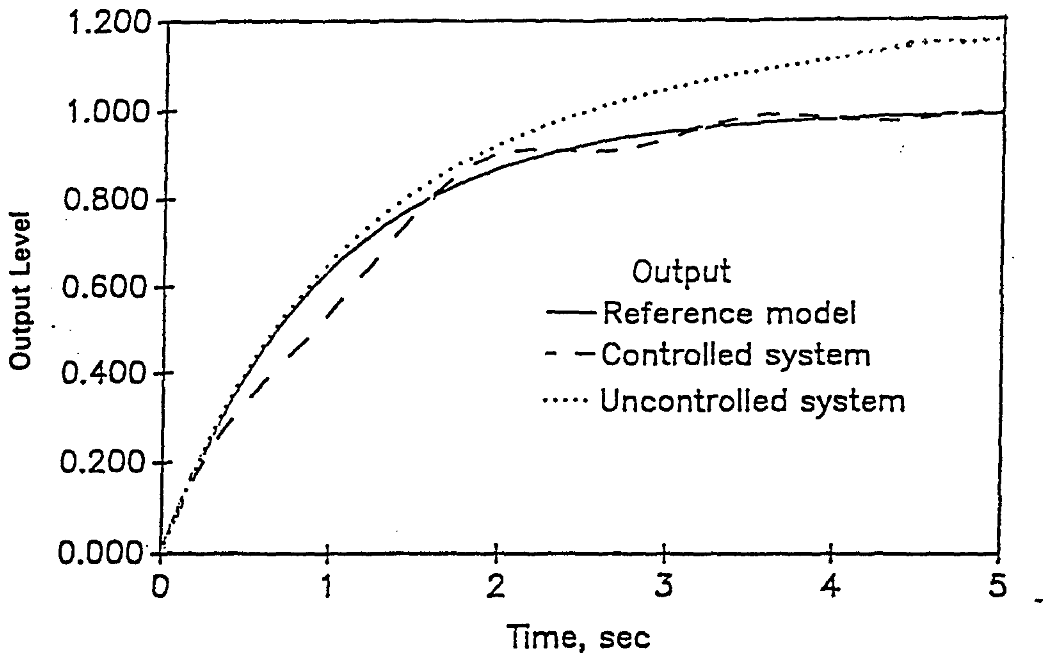


Figure 7-7. Transient, step responses of a time-invariant reference model and a time varying dynamic system.

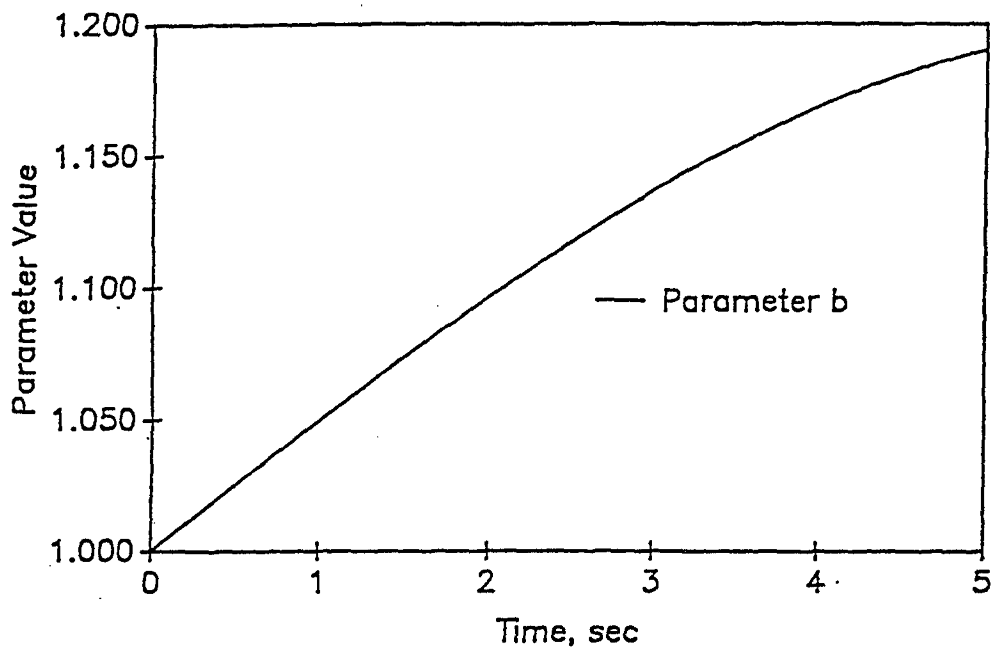


Figure 7-8. Time-varying parameters b.

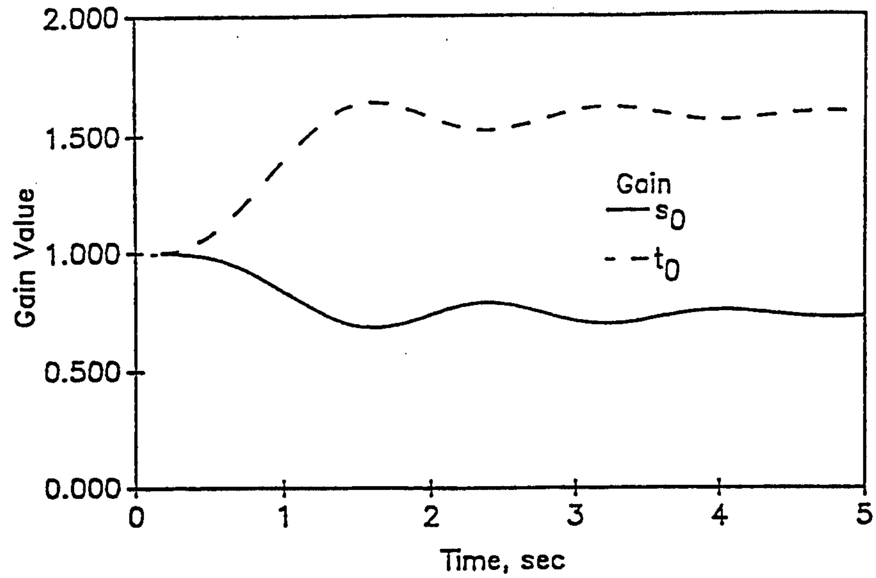


Figure 7-9. Time-varying gains.

This example can be generalized to many situations of interest. The critical assumption is that the uncertain dynamic system has a pole excess of 1, i.e., the dynamic system has one more pole than it has zeros. The results in the more general case take the form of:

$$\frac{d\theta\,(t)}{dt} = \gamma\,\phi\, e\,(t)\ ,$$

with Θ a vector of controller parameters and ϕ a vector of measurable quantities.

7.5 Self-Tuning Regulator Adaptive Control Systems

Many problems involving the control of dynamic systems can be solved by implementing relatively simple controllers based on classical control system design methods. Increased demand for higher system performance and more efficient operation has led to more complex controllers and better tuned systems. A complex regulator design may require feedforward control elements, the use of a state observer, and state variable feedback to accomplish its objectives. A regulator of this complexity for a single-input, single-output linear dynamic system may have as many as ten parameters that must be selected by the control system designer during a tuning process.

The application of more complex regulators has long been hampered by the lack of a systematic process for tuning the control system. The difficulty of this tuning process is one factor which suggests the use of an adaptive control system. Once implemented, the adaptive control system performs the necessary tuning process automatically and continuously without intervention by an operator or control system designer.

Classical control loop design methods are based on mathematical models of dynamic systems and the disturbances which act on them. Often, the precise nature of the dynamic system and the disturbances are unknown, and their mathematical properties must be evaluated. This evaluation process can be done off-line, based on an analysis of collected test data. The evaluation process can also be implemented on-line. In that case the parameters describing both the dynamic system and the disturbances are continuously updated. When the parameters of the dynamic system and the disturbance are known, a suitable controller can be designed to regulate the dynamic system's performance and compensate for external disturbance effects.

7.5.1 Self-Tuning Regulator Design Principles

A self-tuning regulator is a type of adaptive control system used to control a dynamic system having either constant or slowly varying parameters. The design of self-tuning regulators is based on the principle of certainty equivalence. This principle allows the design of the controller and the design of the requisite state estimator to be separated. The most common self-tuning regulator design employs a least-squares estimation process. By combining different state estimators with different controller design methodologies, a wide spectrum of regulator designs can be developed.

A self-tuning regulator can be considered a composite assembly of an on-line system parameter estimator and a continuously implemented control system design procedure. In a self-tuning regulator, the regulator parameters are continuously adjusted in accordance with a specified design procedure. This design procedure requires as an input the latest estimate of the dynamic system state variables or the identified parameters of a mathematical model for the dynamic system.

Figure 7-5 illustrated the basic structure of a self-tuning regulator. The self-tuning regulator consists of two control loops which operate simultaneously. The inner control loop consists of the uncertain dynamic system and an adjustable control device. The control device's parameters are continuously adjusted by the outer control loop which consists of a mechanism for evaluating the current parameters of the dynamic system and a design procedure. Both control loops are usually implemented in software. The structure of the controller depends on the designer's choices for the evaluator and the controller design procedure.

The design of a self-tuning regulator is based on the certainty equivalence principle[7.1] which allows the design of the evaluator and the controller to be separated. The evaluator, either a state variable estimator or a parameter identifier, is first designed and used to construct a model for the dynamic system. This mathematical model is based on measurements of the dynamic system's input and output. The resulting mathematical model is then used as if it were the true model for the dynamic system. Any uncertainties in the model parameters or state variables are ignored, and a suitable controller is designed and implemented to generate a control input signal for the dynamic system.

The self-tuning regulator was originally proposed in 1958 by Kalman[7.8]. Self-tuning regulators have been applied with much success in the process control industries, and have received considerable attention because of their good transient response and asymptotic properties.

If the underlying dynamic system's mathematical structure is unknown, the design of a self-tuning regulator is initiated with the choice of a dynamic system model structure, an order and a delay or sampling time. Since the model's parameters will be evaluated on-line, the model order can be over-stated at the expense of increased computational burden. The delay time must be sufficiently small, yielding a sufficiently high sampling frequency. The self-tuning regulator will be unstable if the dynamic system has a zero in the right-hand plane. This can be overcome by several methods[7.9, 7.10]. For a dynamic system with unknown but near constant parameters, use of a least squares identifier will guarantee convergence to an estimated parameter set.

The self-tuning regulator can also be used in a single-tuning cycle mode. In that case the self-tuning process is turned off after one tuning cycle has been completed. A discount, or forgetting, factor may be applied to the data if the objective is to track and adapt to a set of changing plant dynamics.

The self-tuning regulator can be modeled by the following discrete time system:

$$A\left(z^{-1}\right) y(t) = B\left(z^{-1}\right) u(t-k) + c\left(z^{-1}\right) w(t)$$

where

z^{-1}	=	the delay operator,
A, B, and C	=	polynomials of order n in z^{-1},
k	=	the delay,
y(t)	=	the system output,
u(t)	=	the system input, and
w(t)	=	a zero-mean, Gaussian white noise sequence.

An equivalent representation is:

$$y(t+k) = a_0 y(t) + \dots + a_n y(t-n) +$$

$$b_0\left[u(t) + \dots + b_1 u(t-1) + \dots + b_{n+k} u(t-n-k)\right] + w(t+k)$$

Identification of the system dynamics is usually done by a least-squares method and the resulting control law is:

$$u(t) = \left[\frac{-1}{b_0}\right] [a_1'y(t) + \ldots + a_n'y(t-n)$$

$$- b_1'u(t-1) - \ldots - b_{n+k}'u(t-n-k)]$$

where the apostrophes indicate an estimated parameter. This control law causes the parameter estimates to be unbiased and, under certain technical conditions, to converge to the values of the minimum variance regulator. The self-tuning regulator method can be extended to handle reference inputs, which appear as additional terms on the right- hand side of the control law equation.

7.5.2 Self-Tuning Regulator Design Example

The design of a self-tuning regulator is based on the certainty equivalence principle, a fundamental principle involving the separation of the dynamic system estimation and control functions. The design process begins by selecting an appropriate control system design method for a known dynamic system. Two methods which have received wide attention are the pole-placement method and the method of linear-quadratic design. The problem with applying either of these methods directly is that since the dynamic system is unknown, the parameters or gains of the closed-loop controller cannot be determined. In the self-tuning regulator approach to adaptive control, the parameters of the unknown dynamic system are generated by a recursive parameter estimator which implements a method of system identification. Many techniques for recursive parameter estimation are available and several have been discussed elsewhere in this report.

Figure 7-5 showed the basic configuration of a self-tuning regulator. This type of adaptive control system has three major components: a parameter estimator, a controller, and a design procedure which determines the controller parameters based on the parameter estimates. The following example will illustrate the interaction between these three components.

A dynamic system is governed by the first-order difference equation:

$$y(t) = a\,y(t-1) + b\,u(t-1) ,$$

where

$y(t)$ = the system output at time t, and
$u(t)$ = the system input at time t.

The output y(t) is to be regulated about the state y(t) = 0. The response of this system to an initial displacement is determined by the pole located at z = a. The speed of response can be increased by shifting the pole location to higher value a′. In this example the pole is originally at z = 0.5 and the parameter b equals 1.0. The desired value of a′ will be taken to be 0.9. This can be accomplished by applying the control input:

$$u(t-1) = \left[\frac{1}{b}\right] (a' - a)\, y(t-1) \; .$$

This result, which can be obtained algebraically, assumes that the parameters a and b of the dynamic system are known. When these parameters are uncertain, a self-tuning regulator can be applied to provide adaptive regulation of the output y(t). A self-tuning regulator can be obtained by combining estimation of the unknown parameters (equivalent to identification of the unknown dynamic system) with the development of an appropriate control law. The parameters a and b of the dynamic system will be estimated by a recursive estimator. The resulting parameter estimates are a″(t) and b″(t). The control input:

$$u(t-1) = \left[\frac{1}{b''(t-1)}\right] \left(a' - a''(t-1)\right) y(t-1)$$

will then be applied to the system. This input is computed based on the estimated system parameters.

Any one of several recursive parameter estimation techniques, including the least-squares method, can be applied to develop the parameter estimates and provide an identification of the dynamic system in terms of a″(t) and b″(t). As an example, the following gradient algorithm will be used:

$$p''(t) = p''(t-1) + (\mathrm{norm}^2\left(x(t)\right)^{-1} x(t)\, e(t) \; ,$$

where

$$\begin{aligned} p''(t) &= [a''(t), b''(t)]^T, \\ x(t) &= [y(t-1), u(t-1)]^T, \text{ and} \\ e(t) &= y(t) - [a''(t-1)\, y(t-1) + b''(t-1)\, u(t-1)]. \end{aligned}$$

This method is called a pole-placement self-tuning regulator. Figure 7-10 shows the way in which the estimated parameters a″ and b″ evolve over time. Figure 7-11 shows the response of the controlled system, the response of the uncontrolled system, and the control input. Note that the controlled system output is driven to the origin much faster than the response of the uncontrolled system. Also note that the parameter estimates do not reach the precise values of the underlying dynamic system. The reason for this is the gradient algorithm. As the error e(t) approaches zero, the parameter updates change more slowly. In this example, the error is rapidly driven to zero before the parameter estimates converge to the true parameter values.

7.6 Comparing Model Reference and Self-Tuning Adaptive Control Systems

Recent research and development of adaptive control systems has been focused on two basic approaches: the model reference adaptive control system and the self-tuning regulator. Both self-tuning regulators and model reference control systems have been successfully applied to the design of adaptive control systems. A comparison of the features of these two methods will exhibit their similarities and differences.

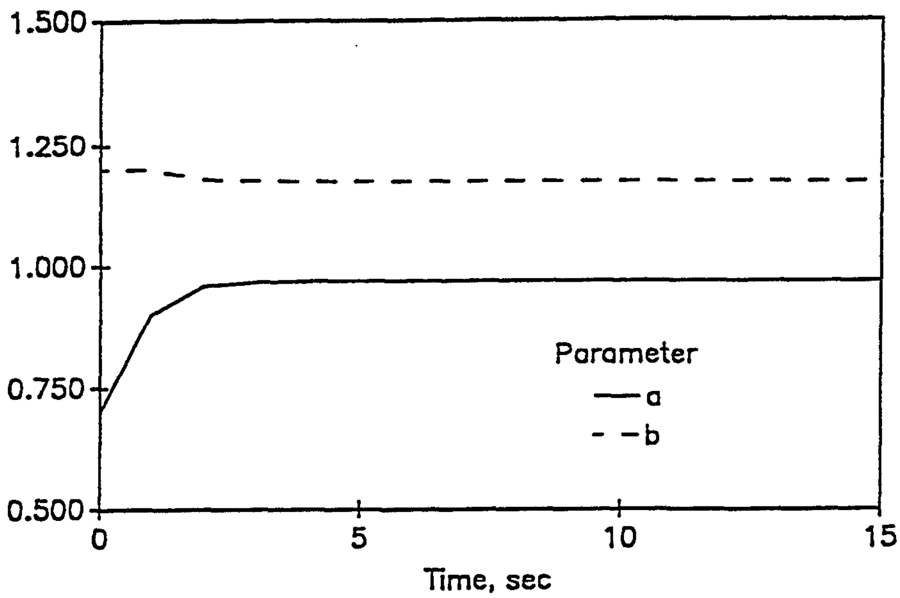


Figure 7-10. Parameter estimates for a self-tuning regulator.

The design of a model reference adaptive control system assumes that the desired dynamic system response can be specified in terms of a reference model. The term model-following control system is also used to describe this concept, and emphasizes the fact that the model is selected in advance and produces a specific desired output when a known command signal is input. This is a typical servo problem, and the model reference adaptive control system method has primarily been applied to adaptive servo problems.

Most model reference adaptive control system design and analysis is done assuming that the underlying dynamic system is deterministic. The previous example illustrated the design of a specific model reference control system for a first-order continuous time dynamic system. For simulation purposes, the continuous-time model was converted to a discrete-time model by means of rectangular integration. Most adaptive control work done today relies on computer implementations and as a consequence, the techniques of discrete-time systems are used.

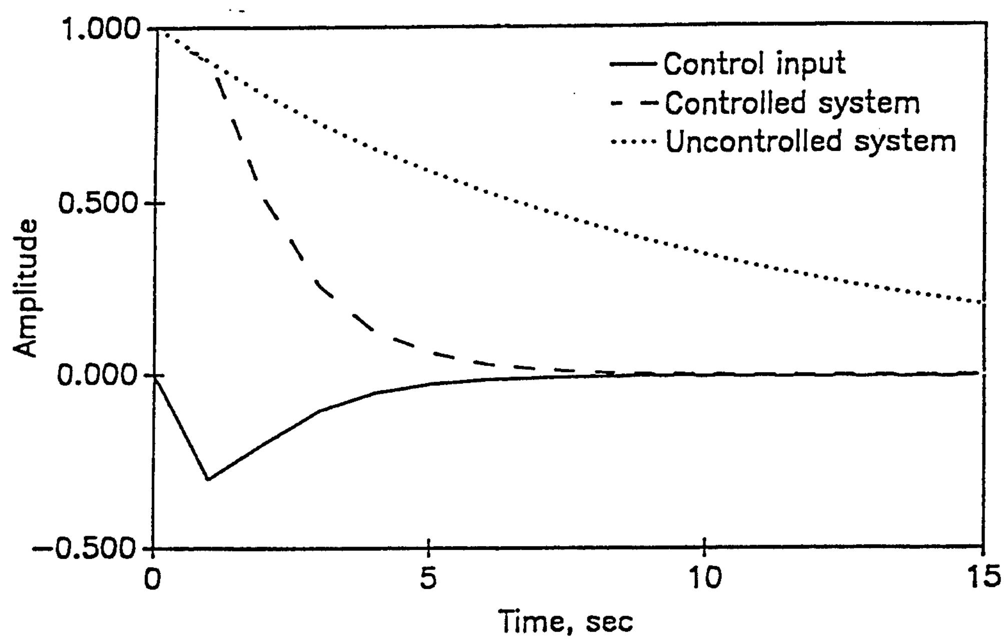


Figure 7-11. Transient response and control input for self-tuning regulator.

The self-tuning regulator concept allows a very general class of control problems to be considered. Any control system design method developed for a known class of dynamic systems having known system parameters can be coupled with a selected recursive parameter estimation algorithm to construct a self-tuning regulator.

The general principle underlying the operation of a self-tuning regulator does not give any hint as to how the processes of estimation and control should be combined. Occasionally, a change of parameters will yield a simpler relationship between the estimated system parameters and the parameters of the appropriate controller. When this is done, the resulting adaptive control system can closely resemble a model reference control system. The condition that the linear model of the underlying dynamic system be minimum phase, possessing no zeros in the right-hand plane, applies to both types of adaptive systems.

The design of an effective self-tuning regulator relies on the design and application of system identification methods. Identification algorithms, such as the recursive least-squares method, can be used in a self-tuning regulator design without major modification.

The design of model reference control systems is based on the principles and techniques of stability analysis. This approach leads to a systematic design procedure for linear dynamic systems having a pole excess of one. Various technical devices have been introduced to extend the application of model reference control systems to linear dynamic systems having pole excess greater than one[7.11].

The distinction between model reference adaptive control systems and self-tuning regulators is not always clear. Model reference adaptive control systems are primarily applied to deterministic control problems in which the parameters of the controller are directly adjusted. Landau[7.12] has attempted to combine these two design concepts in a unified framework.

7.7 Summary

This chapter has outlined the fundamental concepts of adaptive control. The basic structure of an adaptive control system was presented, and a comparison of classical feedback control and adaptive control was made. The differences between gain scheduling, model reference adaptive control, and the self-tuning regulator were discussed, and several examples were presented to illustrate these concepts.

Theoretical progress in the area of adaptive control systems has been rather slow, and design methodologies are often heuristic. The closed-loop control system obtained from a design based on adaptive control technology is more complex than that based on classical control system design methods, and, as a consequence, is difficult to analyze in detail. The problem is compounded if random disturbances are present.

The stability of the overall closed-loop system cannot generally be guaranteed, convergence rates are usually not predictable, and modeling errors can affect the overall performance. The transient response of classical continuous-time and discrete-time control systems is well-understood, and analytic methods are available to predict and control the transient response of simple dynamic systems. Very little is presently known about the transient response of adaptive control systems. An asymptotic theory has been developed, but this theory has strong technical restrictions. Continuous-time dynamic systems for which an adaptive controller is desired may not have any transfer function zeros in the right-hand complex plane. To fully apply existing asymptotic theory, the dynamic process must have a known delay if a discrete-time system, a known structure if a multi-variable system, and no unmodeled dynamics.

The present application of adaptive control technology requires substantial knowledge of the underlying system dynamics, considerable ingenuity, and a willingness to undertake an extensive simulation effort. Many open questions regarding the design and application of adaptive control systems remain and are the subject of considerable ongoing research. Despite these largely theoretical drawbacks, progress in the application of adaptive control theory has been rapid in recent years. Most applications of adaptive control theory have, with few exceptions, not been in aerospace or missile guidance and control, but the potential is great and more applications are expected.

REFERENCES

7.1 Astrom, K.J. and Wittenmark, B., Adaptive Control, Addison-Wesley Publishing company, Reading, Massachusetts, 1989.

7.2 Stein, G., Adaptive Flight Control - A Pragmatic Approach in Applications of Adaptive Control, Narenda, K.S. and Monopoli, R.V., eds., Academic Press, New York 1908.

7.3 Astrom, K.J., Theory and Applications of Adaptive Control, Automatica, Vol. 19, pp. 471-486, 1983.

7.4 Isermann, R., Parametric Adaptive Control Algorithms - A Tutorial, Automatica, Vol. 18, pp. 513-528, 1982.

7.5 Landau, Y.D., Adaptive Control - The Model Reference Approach, Marcel Dekker, New York, 1979.

7.6 Landau, Y.D., A Survey of Model Reference Adaptive Techniques (Theory and Application), Automatica, Vol.10, pp.353-379, 1974.

7.7 Parks, P.C., Lyapunov Redesign of Model Reference Adaptive Control Systems, IEEE Transactions on Automatic Control, Vol. TAC-11,pp. 362-365, 1966.

7.8 Kalman, R.E., Design of a Self-Optimizing Control System, Transactions of the ASME, Vol. 80, pp. 468-478, 1958.

7.9 Clarke, D.W. and Gawthrop, P.J., A Self-Tuning Controller, IEEE Proceedings, Vol. 122, pp. 929-934, 1975.

7.10 Wittenmark, B., A Self-Tuning Regulator, Ph.D., Thesis CODEN:LUTFD2/TFRT-1003, Department of Automatic Control, Lund Institute of technology, Lund, Sweden, 1973.

7.11 Monopoli, R.V., Model Reference Automatic Control With An Augmented Error Signal, IEEE Transactions on Automatic Control, Vol. TAC-19, pp. 474-484, 1974.

7.12 Landau, I.D., Model Reference Adaptive Controllers and Stochastic Self-Tuning Regulators - A Unified Approach, Journal of Dynamic Systems, Measurement and Control, Vol. 103, pp. 404-416, 1981.

This page is intentionally blank.

CHAPTER 8
MATHEMATICAL OPTIMIZATION

8.1 Static Optimization Problem

Decision problems involving the best numerical values assigned to control system parameters, the best trajectory followed by a missile enroute to its target, or the best input signal applied to drive a dynamic system to some desired state are all problems of mathematical optimization. Optimization theory and the development of algorithms for mathematical optimization are an important segment of modern control theory. Optimization problems can be classified in many ways, and in this chapter the fundamental problem structures and concepts of mathematical optimization are outlined.

A static, non-time-dependent optimization problem involves minimizing a function of a set of variables, where the variables are restricted according to a set of constraints. The variables in question may be real valued or may take on only integer values. The function to be minimized is generally not restricted in form. Some functions, such as a linear or quadratic combination of variables, lend themselves to the development of problem-specific solution algorithms which have widespread application.

Mathematically, a static optimization problem is described by a problem statement having the following form:

$$\text{minimize} \quad f(x), \; x \in C ,$$

where

C = the constant set.

The function f(.) is called the objective, cost, or payoff function. The objective function mathematically models the result of choosing to use a particular value of x. The constraint set C consists of a set of equalities, inequalities, or other mathematical relationships which define or restrict the values or range of values of x which are considered acceptable, or feasible. By defining the constraint set C and the function f(.), a wide range of optimization, design, and control problems can be formulated and solved.

If the constraint set consists of a set of nonlinear inequalities or equalities having the form:

$$g_i(x) \leq 0, \; i = 1, 2, \ldots, L ,$$

the problem is called a nonlinear programming problem. If the objective function is a linear weighted function of the variables x, and the constraint set C consists of a set of linear inequalities or equalities having the form:

$$\sum_{i=1}^{i=N} a_i \, x_i \leq 0, \; i = 1, 2, \ldots, L \; ,$$

the problem is called a linear programming problem. A sophisticated, high-speed algorithm, the Simplex method, exists to solve linear programming problems. If the variables are restricted to have integer values instead of real values, the problem is said to be an integer programming problem.

Static optimization problems usually involve problems having a finite, or limited, number of variables. To extend these optimization concepts to problems having an infinite number of variables, additional theoretical notions are required. For example, a function of time, x(t), represents a quantity that takes on some value at each instant of time. A function of this sort involves, in effect, an infinite number of variables, one at each time instant. Infinite sequences of variables are another example of a set of numbers containing an infinite number of items.

The solution of optimization problems for infinite-dimensional variables is the province of the calculus of variations and dynamic optimization and control theory. Dynamic optimization is a key area of research and applications in modern control theory. Dynamic optimization involves finding solutions to optimization problems in which the answer, or solution, is a function of time, rather than a set of numerical values for the variables x as in a static optimization problem.

There are three areas of concern in the study of static optimization problems:

(1) whether or not a solution to a particular problem exists, and, if a solution exists, whether or not it is unique,

(2) what the necessary conditions are for an optimal solution, i.e., if a solution is optimal, what set of conditions does it satisfy, and

(3) what numerical algorithms are available or can be developed to find the optimal solution.

If it can be determined in advance that a solution does exist for a particular static optimization problem, then the search for that solution and the development of numerical algorithms may be worth the effort and expense. The necessary conditions provide a means to test potential optimal solutions.

For example, a continuous function f(x) will have either a local maximum or minimum at a point where the slope of the function, or the derivative, is zero. This necessary condition allows potential optimal solutions to be identified, but does not provide any information about whether or not a point at which the derivative is zero is a maximum or minimum. Additional information is usually needed to solve the optimization problem.

Certain problem types, such as the linear programming problem, are characterized by well-defined numerical algorithms which can solve problems of arbitrary complexity and indicate whether or not a solution exists, and if a solution exists, whether or not there are alternate optimal solutions which yield the same value of the objective function.

8.2 Linear Programming

The mathematical programming technique known as linear programming was developed in the late 1940s by George Dantzig. Linear programming[8.1,8.2] is today the most widely used numerical optimization technique, but its application to problems arising in the area of modern control theory has only recently been realized. This technique also forms a basis for more complicated numerical algorithms in which a sequence of linear programming problems is solved to obtain the solution to a nonlinear optimization problem.

Certain problems involving the optimal control of discrete-time dynamic systems can be solved by casting these problems in the form of a linear programming problem.

The standard mathematical model for a linear programming problem is:

$$\text{minimize } c_1 x_1 + c_2 x_2 + \dots + c_n x_n$$

subject to:

$$a_{11} x_1 + a_{12} x_2 + \dots + a_{1n} x_n = b_1$$

$$a_{21} x_1 + a_{22} x_2 + \dots + a_{2n} x_n = b_2$$

$$\dots$$

$$a_{m1} x_1 + a_{m2} x_2 + \dots + a_{mn} x_n = b_m$$

$$x_i \geq 0,\ i = 1, 2, \dots, n$$

$$b_j \geq 0,\ j = 1, 2, \dots, m\ .$$

This problem has n unknown variables (x_1, x_2, ..., x_n), and m linear equalities which serve as constraints which describe the relationships between these unknown variables. Linear programming problems having other forms can be converted to this standard form by adding so-called slack or surplus variables used to convert inequalities to equalities.

The general linear programming problem can be compactly written in matrix form as:

minimize $C^T x$

subject to:

$$\begin{aligned} Ax &= b \\ x &\geq 0 \,. \end{aligned}$$

In this notation x is an n-dimensional vector $[x_1, x_2, \ldots, x_n]$ of unknown variables, the vector c is an n-dimensional vector of costs, A is an n by m matrix of constants, and b is an m-dimensional vector of positive values.

To illustrate how linear programming can be applied to solve certain optimal control problems an example will be used. A dynamic system is described by the differential equation:

$$\frac{dx(t)}{dt} = -ax(t) + u(t) \,,$$

with $x(0) = x_0$ a specified initial condition and

$$0 \leq u(t) \leq u_{max} \,.$$

The performance measure to be minimized is:

$$J = \int_{t=0}^{t=T} g(t) \left[x_d(t) - x(t)\right] dt$$

and the system output x(t) is to be constrained so that it is less than or equal to the desired output $x_d(t)$ for each value of time t in the range from 0 to T, the final time of the problem. The function g(t) is a weighting function which is used by the control system designer to assign a relative importance to the response at each time instant. This problem is called a curve-tracking problem.

The first step is to convert this continuous-time problem to a discrete-time problem suitable for computer solution and implementation. This is done by converting the differential equation to a difference equation and converting the performance measure from an integral to a finite sum:

$$x(k+1) - x(k) = \left[\frac{T}{K}\right] \left(-ax(k) + u(k)\right), \; k = 0, 1, \ldots, K - 1 \,,$$

$$x(0) = x_0 \,.$$

Re-writing this equation one obtains:

$$x(k+1) = \left[1 - \left[\frac{T_a}{K}\right]\right] x(k) + \left[\frac{T}{K}\right] u(k),\ k = 0, 1, \ldots, K - 1 ,$$

$$x(0) = x_0 .$$

This is a set of K + 1 equalities in terms of the variables x and u. To further simplify the notation let $p = (1-(T_a/K))$ and $q = (T/K)$:

$$x(k+1) = p\,x(k) + q\,u(k),\ k = 0, 1, \ldots, K - 1 ,$$

$$x(0) = x_0 .$$

Next write these equalities in recursive form, substituting the prior value of x(k) in each:

$$x(0) = x_0$$

$$x(1) = p\,x_0 + q\,u(0)$$

$$x(2) = p^2 x_0 + p\,q\,u(0) + q\,u(1)$$

$$x(3) = p^3 x_0 + p^2 q\,u(0) + p\,q\,u(1) + q\,u(2) .$$

...

$$x(K) = p^K x_0 + \sum_{j=0}^{j=k-1} p^{K-1-j} u(j)$$

These equalities give the sequence of states, x(k), in terms of the control sequence, u(k).

The performance measure can be converted to a summation directly:

$$J = \sum_{k=1}^{k=K} g(k) \left(x_d(k) - x(k)\right) q .$$

Finally, the equalities can be converted to inequalities in terms of the control variables by the addition of the specified constraints, slack variables, and the collection of constant terms:

$$q\,u(0) \qquad\qquad\qquad + s(1) = x_d(1) - p^1 x_0$$

$$p\,q\,u(0) + q\,u(1) + q\,u(2) + s(2) = x_d(2) - p^2 x_0$$

$$p^2 q\,u(0) + p\,q\,u(1) + q\,u(2) + s(3) = x_d(3) - p^3 x_0$$

...

$$q \sum_{k=1}^{k=K-1} p^{K-1-j}\, u(j) + s(K) = x_d(k) - p^K x_0 \; .$$

The curve tracking problem is now nearly in the form of a standard linear programming problem:

$$\text{minimize } J = \sum_{k=1}^{k=K} g(k) \left(x_d(k) - x(k)\right) q$$

subject to:

$$x\,0) = x_0$$

$$x(1) = p x_0 + q u(0)$$

$$x(2) = p^2 x_0 + p q u(0) + q u(1)$$

$$x(3) = p^3 x_0 + p^2 q u(0) + p q u(1) + q u(2)$$

...

$$x(K) = p^K x_0 + q \sum_{j=0}^{j=K-1} p^{K-1-j}\, u(j)$$

$$q u(0) \qquad\qquad\qquad + s(1) = x_d(1) - p^1 x_0$$

$$p q u(0) \;+ q u(1) \qquad\quad + s(2) = x_d(2) - p^2 x_0$$

$$p^2 q u(0) + p q u(1) + q u(2) + s(3) = x_d(3) - p^3 x_0$$

...

$$q \sum_{k=1}^{k=K-1} p^{K-1-j}\, u(j) + s(K) = x_d(k) - p^K x_0 \; .$$

$$0 \le u(j) \le u_{max},\ j = 1, 2, \ldots, K-1 \; ,$$

$$0 \le x(j),\ j = 1, 2, \ldots, K \; .$$

To place this problem in standard form, the right-hand side of each equality must be made a positive constant. The Simplex algorithm can then be applied to solve this mathematical optimization problem and obtain the optimal control sequence u and the sequence of states x. The state sequence could be eliminated by further substitution, thus reducing the number of problem variables. This was not done in this example in the interest of retaining the dynamic nature of the system and illustrating how the linear programming problem generates as a solution both the control sequence and the state sequence.

By modifying the performance measure, the dynamic equations and the constraints of this problem, other problem structures can be developed which permit linear programming to solve minimum time, minimum fuel, or other similar dynamic optimization problems.

8.3 Nonlinear Programming

Nonlinear programming problems are static optimization problems which involve the maximization or minimization of a function of one or more variables, where either the function or the constraints are nonlinear in terms of the variables. Many optimal control problems in modern control theory can be cast as nonlinear programming problems. The process of system identification and the estimation of unknown system parameters both involve the solution of nonlinear optimization problems. The method of least squares, for example, requires that a quadratic function of several variables be minimized. The solution sought is that combination of variables which yields a minimum of the quadratic function.

Solution methods and algorithms for nonlinear programming problems which are as well developed as the Simplex method for solving linear programming problems are rare. Most algorithms are ad hoc, problem-specific techniques not easily transferable from one nonlinear problem to another.

Unconstrained nonlinear programming problems, in which the function to be optimized is nonlinear but the variables are not constrained in any way, can usually be solved by one or more of these standard methods:

(1) gradient descent method,
(2) conjugate gradient method,
(3) Newton's method, or
(4) Quasi-Newton methods.

If the constraint set is linear in terms of the problem variables but the performance measure is a quadratic function, the method of quadratic programming can be applied. Random search methods can also be effective, especially when the number of problem variables are small.

8.3.1 Gradient Descent Method

The nonlinear programming problem to be solved is written in the form:

minimize $z = f(x)$,

where

$$[x_1, x_2, \ldots, x_n]^T .$$

There are no constraints on the magnitudes of the components of the vector x.

The gradient descent method, also called the method of steepest descent, can be summarized in the following algorithm:

(1) Choose an initial vector x_0, using any prior information about the location of the optimal solution which might be available.

(2) Determine successive vectors x^1, x^2, ... by the recursive formula:

$$x^{k+1} = x^k + \lambda^k \frac{\partial f}{\partial x^k}$$

where

$$\frac{\partial f}{\partial x^k} \text{ is the vector}$$

$$\left[\frac{\partial f}{\partial x^1}, \frac{\partial f}{\partial x^2}, \ldots, \frac{\partial f}{\partial x^n}\right]^T$$

evaluated at x^k and λ^k is a positive scalar which minimizes:

$$f\left\{x^k + \lambda^k \frac{\partial f}{\partial x^k}\right\} .$$

The single variable optimization problem involving λ can be solved by any one of the following sequential-search techniques:

(a) three-point interval search,
(b) Fibonacci search, or
(c) Golden-mean search.

(3) Terminate the recursive process if and when the difference between any two successive vectors x^k and x^{k-1} is less than a pre-determined tolerance.

<u>Three-Point Interval Search</u>. In the three-point interval search the interval being considered,

$$x^k + \lambda^k \frac{\partial f}{\partial x^k}$$

is divided into quarters and the function f(.) evaluated at each of the three equally spaced points within the interval. The point yielding the minimum value of f(.) is determined, and the sub-interval centered on this point and made up of one-half the present interval becomes the next-current interval.

The process is repeated until the minimum value of f(.) is found to within a pre-determined tolerance. The three-point interval search is easily implemented in software and is the most efficient equally spaced search technique. This technique can achieve a solution to within a pre-determined tolerance utilizing a minimum number of function evaluations.

Fibonacci Search. The Fibonacci search technique is the most efficient of all sequential search methods. The Fibonacci sequence, in which the first two numbers are both one and the successive numbers are the sum of the previous two numbers, forms the basic structure of this method. The Fibonacci search is started by determining the smallest Fibonacci number F_n which satisfies the inequality $F_n \cdot \epsilon \geq (b-a)$, where ϵ is a pre-determined tolerance and a and b are the endpoints of the search interval. The Fibonacci sequence is $F_n = [1, 1, 2, 3, 5, 8, 13, \ldots]$.

A new tolerance $\epsilon' = (a-b)/F_n$ is then determined, and the first two search points are then placed $F_{n-1}\epsilon'$ units in from the points a and b. The function f(.) is evaluated at both search points, and that point yielding the minimum becomes the new left or right endpoint. Successive search points are positioned $F_j\epsilon$ units in from the endpoints of the current interval.

The advantage of the Fibonacci search technique is that the number of function evaluations required to obtain a pre-determined tolerance can be determined in advance, and that number is independent of the function being evaluated.

Golden-Mean Search. The number [sqrt(5) − 1]/2 = 0.6180 ... is called the golden mean. In the golden-mean sequential search technique the first two search points are located $0.6180 \cdot (b-a)$ in from the left and right endpoints a and b. Successive endpoints are positioned at 0.6180 L units in from the newest endpoints of the current interval L.

8.3.2 Conjugate Gradient Method

The gradient descent method outlined above searches for the minimum of the function f(.) in a steepest descent direction. Movement from one potential solution to another is along the direction of the gradient of the function. This method often yields an oscillatory approach to the local optimum, especially when the contours of the objective function in the neighborhood of the optimum are elongated. Other search methods, which use the direction of the gradient as a means for determining the direction of motion but do not actually move along that direction when seeking an improved solution, have been developed.

The conjugate gradient method converges to the optimal solution in a quadratic manner, and also finds the minimum of the quadratic objective function:

$$z = f(x^*) + \left(\frac{1}{2}\right) (x-x^*)^T A (x-x^*) ,$$

where A equals an n by n positive-definite matrix in a finite number of iterations, usually equal to n. The conjugate gradient method and similar gradient methods search in the direction of the conjugate gradient, and guarantee that the optimum is found within a finite number of iterations. The Fletcher-Reeves algorithm is one implementation of a conjugate gradient sequential search technique.

8.3.3 Newton's Method

Newton's method, also called the Newton-Raphson method, is an easily implemented sequential search technique. An initial vector x^1 is selected as in the gradient descent method. Successive vectors x^1, x^2, ... are recursively determined by the algorithm:

$$x^{k+1} = x^k - \left[H_f(x^k)\right]^{-1} \frac{\partial f}{\partial x^k}$$

where

$$\frac{\partial f}{\partial x^k} = \left[\frac{\partial f}{\partial x^1}, \frac{\partial f}{\partial x^2}, \ldots, \frac{\partial f}{\partial x^n}\right]^T$$

evaluated at x^K and

$$H_f(x^k) = \left[\frac{\partial^2 f}{\partial x^i\, \partial x^j}\right], \; i, j = 1, 2, \ldots, n$$

is a Hessian matrix of second partial derivatives. The iterations are stopped whenever two successive vectors are equal to within a pre-determined tolerance.

8.3.4 Quasi-Newton Methods

Although Newton's method exhibits good convergence to the optimal solution x^* in the neighborhood of x^*, each step requires the evaluation of $n(n+1)/2$ second-order partial derivatives of f(.) to determine the Hessian matrix and the inverse of an n by n matrix which involves approximately n^3 multiplications. The method must also be modified for application to objective functions which are not convex, since the inverse matrix required may not exist.

Quasi-Newton methods involve the use of approximations to generate the inverse Hessian matrix. In general these methods implement a recursive relation of the form:

$$x^{k+1} = x^k + a^k \delta^k ,$$

where

$$\delta^k = -H^k \partial \frac{f}{\partial x^k},$$

and

H^k = a positive-definite symmetric matrix.

The initial matrix H^0 is arbitrary, and an identity matrix is conveniently used. The matrix H^k is updated at each iteration such that the method approximates Newton's method. The step size α^k is also selected at each iteration.

8.4 Calculus of Variations

The calculus of variations is a branch of optimization theory which involves problems in which the unknown is not a variable, x, but a function, f(x). This function may itself be multi-dimensional. The historical development of the calculus of variations parallels the development of calculus and differential equations. Certain problems in the calculus of variations, in particular finding the area and shape of the largest surface which could be enclosed with a given perimeter, are known to have been studied in ancient times. Since the underlying mathematics of the calculus of variations is somewhat abstract, this material is unfamiliar to most system engineers and designers.

The value of this topic is the fact that many problems in optimal control theory can be formulated as equivalent problems in the calculus of variations, and the mathematics of the calculus of variations forms the basis for optimal control theory.

The classical optimization problem treated in the calculus of variations is called the problem of Lagrange:

$$\text{maximize } J = \int_{t=t_0}^{t=t_f} I \left[x(t), \frac{dx(t)}{dt}, t \right] dt$$

where the initial and final times t_0 and t_f, the initial and final conditions $x(t_0)$ and $x(t_f)$ are specified, and the integrand I[.] is a continuously differentiable function. The objective is to determine the function x(t) which maximizes (or minimizes) the performance measure J. The performance measure in the calculus of variations is called a functional.

From a historic viewpoint, there was much interest during the late seventeenth century in finding the maximum or minimum of certain time-varying quantities. Galileo Galilei first discussed the brachistochrone problem in his Dialogues [1632]. This problem involved finding the shortest time required for a small frictionless bead to slide under gravity's influence from one given point on a wire to a lower point on the wire. By varying the shape of the wire different trajectories and transition times could be obtained. John Bernoulli [1696] challenged the mathematical community to solve this problem, and a solution was developed by James Bernoulli and others. The optimal wire shape was found to be a cycloid curve, and the name brachistochrone was given to the curve of fastest descent.

Euler, a pupil of John Bernoulli, is credited with establishing the calculus of variations as a mathematical discipline. Euler derived a differential equation which plays an essential role in the solution of calculus of variations problems. Lagrange later simplified and generalized Euler's work and the resulting differential equation is now called the Euler-Lagrange equation.

The development of the calculus of variations and its applications in a wide variety of technical areas is detailed in many sources. In what follows, the basic concepts of the calculus of variation are introduced and an example is presented to indicate the material's application.

As a special case, let there be only a single state variable x(t) and let both the initial and terminal points $x(t_0)$ and $x(t_f)$ as well as the time values t_0 and t_f be given. The optimization problem to be solved is:

$$\text{maximize } J = \int_{t=t_0}^{t=t_f} I\left[x(t), \frac{dx(t)}{dt}, t\right] dt$$

where the integrand I[x(t), dx(t)/dt, t] is specified in any particular application. The Euler-Lagrange equation is:

$$\frac{\partial I}{\partial x} - \frac{\partial}{\partial t}\frac{\partial I}{\partial x'},$$

where x' denotes the time derivative dx(t)/dt. This can also be written in expanded form as:

$$\frac{\partial}{\partial x} I\big(x(t), x'(t), t\big) - \frac{\partial}{\partial t}\frac{\partial}{\partial x'} I\big(x(t), x'(t), t\big) = 0 .$$

This equation is assumed to be valid for all time t from t_0 to t_f. Any specified function I[x(t), dx(t)/dt, t] can be partially differentiated with respect to x(t) and $x'(t)$ to yield the necessary terms in the Euler equation, and the second term can then be differentiated with respect to t. The result of these rather cumbersome mathematical operations is an ordinary differential equation, usually of second-order, which may involve products or powers of $x''(t)$, $x'(t)$ and x(t), in which case the differential equation is highly nonlinear, and the presence of the argument t indicates that the coefficients of that equation may also be time-varying.

The Euler-Lagrange equation for the problem posed is thus a nonlinear, ordinary, time-varying, hard to solve second-order differential equation whose solution yields a function x(t). A natural approach toward obtaining the solution is to use numerical integration. However, the two boundary conditions for this second-order differential equation are split. Rather than having $x(t_0)$ and $x'(t_0)$ specified, the available boundary conditions are $x(t_0)$ and $x(t_f)$. To apply numerical integration directly, values for the two boundary conditions, $x(t)$ and $x'(t)$ at either t_0 or t_f are required.

The solution of the Euler-Lagrange differential equation is thus complicated due to the combination of split boundary conditions and nonlinearity of the second-order differential equation. Only very simple problems can be solved analytically to yield solutions in concise form:

(1) If the integrand depends only on $x'(t)$ the Euler-Lagrange equation reduces to:

$$\frac{\partial^2}{\partial x^2} I\left(x'(t)\right) x''(t) = 0 .$$

In this case, either $\frac{\partial^2}{\partial x^2} = 0$, or $x''(y) = 0$.

$$\text{if } x''(t) = 0, \text{ then } x(t) = c_1 t + c_2 .$$

If the second partial derivative factor equals zero and has a real root $x'(t) = c_3$, then

$$x(t) = c_3 t + c_4 .$$

In either case, two constants of integration are involved and the solution is a family of straight lines.

(2) If the integrand depends only on $x'(t)$ and t, the Euler-Lagrange equation reduces to:

$$\frac{\partial}{\partial t} \frac{\partial}{\partial x'} I\left(x'(t), t\right) = 0 ,$$

which implies that

$$\frac{\partial}{\partial x'} I\left(x(t), t\right) = c_1 .$$

This is a first-order nonlinear differential equation involving $x'(t)$ and t. The required function $x(t)$ can be obtained by solving for $x'(t)$ and then integrating to obtain the function $x(t)$. Two constants of integration are involved.

(3) If the integrand depends only on $x(t)$ and $x'(t)$ the Euler-Lagrange equation reduces to:

$$I\left(x(t), x'(t)\right) = x'(t) \frac{\partial}{\partial x'} I\left(x(t), x'(t)\right) + c_1 .$$

This is also a first-order differential equation involving x'(t). By solving this equation for x'(t) and then integrating x(t) can be determined. Two constants of integration are involved.

(4) If the integrand depends only on x(t) and t the Euler-Lagrange equation reduces to:

$$\frac{\partial}{\partial x} I\left(x(t),\, t\right) = 0 \;.$$

This is a nonlinear algebraic equation involving x(t). No constants of integration are required. The resulting function x(t) is an optimal solution only if the curve described by x(t) passes through the specified boundary points.

(5) If the integrand depends linearly on x'(t) the Euler-Lagrange equation can be written in the form:

$$\frac{\partial}{\partial x} M\left(x(t),\, t\right) - \frac{\partial}{\partial t} N\left(x(t),\, t\right) = 0 \;.$$

This is a nonlinear algebraic equation involving x(t). No constants of integration are required. The resulting function x(t) is an optimal solution only if the curve described by x(t) passes through the specified boundary points.

Example: Use of the Euler-Lagrange Equation. As an example of the application of the calculus of variations to a familiar problem, consider finding the trajectory defined by the function x(t) which begins at $x(t_0) = 1$ when $t_0 = 0$, ends at $x(t_f) = 2$ when $t_f = 2$, and has a minimum length connecting the two points. The increment of arc length along such a trajectory is given by $(1+x'(t)^2)^{1/2}$, and the total arc length can be obtained by summing, or integrating, all of these increments. Since a minimum arc length is desired, the functional will be multiplied by -1 and the maximum of the result determined. The problem to be solved is:

$$\text{maximize } J(x) = \int_{t=t_0}^{t=t_f} \sqrt{1\, x'(t)^2}\; dt,$$

with $t_0 = 0,\; t_f = 1,\; x(t_0) = 1, \text{ and } x(t_f) = 2 \;.$

The integrand $I[(x(t), x'(t), t) = (1+x'(t)^2)^{1/2}$ depends only on x'(t) and so the Euler-Lagrange equation simplifies to:

$$\frac{\partial^2}{\partial x^2} I\left(x'(t)\right) x''(t) = 0,$$

and, either

$$\frac{\partial^2}{\partial x^2} I\left(x'(t)\right) = 0 \text{ or } x''(t) = 0 \;.$$

The solution in either case is a family of straight lines:

$$x(t) = c_1 t + c_2 ,$$

and the split boundary conditions are used to evaluate the two constants of integration:

$$x(t_0) = 1 = c_1(0) + c_2, \text{ or } c_2 = 1, \text{ and}$$

$$x(t_f) = 2 = c_1(1) + 1, \text{ or } c_1 = 1 .$$

The solution to the Euler-Lagrange equation is:

$$x^*(t) = t + 1,\ 0 \le t \le 1, \text{ and } x'(t) = 1 .$$

The value of the functional $J(x^*)$ is:

$$J(x^*) = \int_{t=0}^{t=1} -\left(1+(1)^2\right)^{1/2} dt = -\sqrt{2} .$$

This is the negative value of the total arc length of the trajectory having a minimum arc length and connecting the specified initial and terminal points. The optimal solution is of course a straight line between the two points.

In the discussion to this point the problems considered have had fixed specified endpoints. In more general problems the endpoints may be free or allowed to lie on some terminal surface. In that case additional necessary conditions must be developed. The development of the calculus of variations can also be extended to problems having a multi-dimensional state vector x(t) and constraints on the state x(t) and its derivative x′(t). The interested reader is referred to Kirk[8.3] and Koo[8.4] for a thorough discussion of these various conditions, referred to as the Weierstrauss-Erdmann corner conditions, the Legendre-Clebsch conditions, and the transversality conditions.

Mathematical optimization problems formulated via the calculus of variations nearly always require the solution of a highly nonlinear time-varying multi-dimensional differential equation having split boundary conditions. Such problems cannot generally be solved analytically, and numerical methods for the dynamic optimization must be developed and applied to produce approximate solutions to such problems. For problems having only one or two state variables, the method of dynamic programming, discussed in the next Section, can be applied to produce a solution.

8.5 Dynamic Optimization

While static optimization problems are generally concerned with finding a solution to a problem which does not, in general, involve the passage of time, dynamic optimization problems are concerned with problems in which time is a factor. The theoretical basis of dynamic optimization is the calculus of variations, and the main numerical solution technique is dynamic programming.

8.5.1 Dynamic Systems

A dynamic system is a physical system which evolves over time. A dynamic system is characterized by a set of variables, x, which represent the state of the dynamic system at time t. The manner in which the state changes from one instant to the next is governed by a set of state transition equations. These state transition equations are a mathematical model of the dynamic system and are either in the form of a differential equation:

$$\frac{dx}{dt} = f\left(x(t),\ t\right),$$

or a difference equation:

$$x(k+1) = f\left(x(k),\ k\right).$$

The model is thus deterministic and the state transition equation together with the initial state of the system, x(0), is sufficient to determine the future evolution of the dynamic system.

8.5.2 Controlled Systems

The introduction of an external input, u, often makes it possible to control a dynamic system which would otherwise evolve freely. The state transition equation for a controlled dynamic system takes either the form of a differential equation:

$$\frac{dx}{dt} = f\left(x(t),\ u(t),\ t\right),$$

or a difference equation:

$$x(k+1) = f\left(x(k),\ u(k),\ k\right).$$

The model is deterministic if the state transition equation together with a knowledge of the initial state of the system, x(0), and a knowledge of the control input, u, will completely determine the system's' evolution. The particular control input to be applied must be selected by the control system designer according to some objectives. When these objectives are specified mathematically, the problem of choosing an appropriate control input becomes a problem of optimal control.

For example, the designer might decide to select as a performance measure the following cost function:

$$J = \int_{t=0}^{t=T} g\left(x(t),\ u(t),\ t\right) dt + h\left(x(T)\right) .$$

The integral expresses a time-varying function of the state and the applied control. The second term expresses the dependance of the optimal solution on the final state x(T). The final time, the time interval over which control is to be exerted, is T.

Other performance measures are possible. For example, the goal might be to determine a control input which drives the dynamic system from an initial state x(0) to a final state x(T) in a minimum amount of time.

Practical problems of control almost always involve constraints on the state and control variables. For example, if the control input is the deflection of an aerodynamic surface, that deflection might be limited to a finite range of values due to mechanical considerations. Also, if the state represents the position of an object, it might be desirable to have the object avoid certain regions of space.

Sometimes the final state of the system is specified. In other problems the final state is free, or unspecified. Combinations of these boundary conditions can also occur. For example, the miss distance between a missile and its target may be desired to be minimized, but the particular value is unimportant as long as it is small. In the same problem the velocity of the missile at impact is probably of no consequence.

8.5.3 Feedback and Open-Loop Control

The goal in solving an optimal control problem is to find an acceptable control input u(t) which minimizes the performance measure J. The result of this process is usually an open-loop control system, in which the control input is only a function of time and does not depend on the state of the system.

If a control signal of the form u(t) = L(x(t), t) can be found, in which the control input at time t is computed as a function of the state x(t) and possibly the time t, then the control signal can be implemented as a feedback of the dynamic system state. The use of feedback in linear, constant coefficient, time-invariant dynamic systems offers important advantages, including a reduction in sensitivity of the system's performance in the face of parameter variations due to random perturbations or manufacturing tolerances.

8.6 Dynamic Programming

Dynamic programming is a mathematical optimization technique for systems which can be considered to operate over a number of stages or time steps. The method of dynamic programming can be used to determine optimal control laws in table look-up, feedback form for multistage dynamic systems having one or two state variables. A good example of a multistage process is a dynamic system cast in the form of a discrete-time process and described by a state transition equation of the form:

$$x(k+1) = f\left(x(k), u(k), k\right), \; k = 1, 2, \ldots, K \text{ with}$$

$$x(0) = x_0 \text{ specified} .$$

In this model x(k) is the present state of the system, the state at time k, and u(k) is the control input at time k. The performance of the system is measured by the objective function J:

$$J = \sum_{k=0}^{k=K-1} g\left(x(k), u(k), k\right) + h\left(x(K)\right) .$$

This objective function consists of two parts, the first a summation of the accumulated costs over the K stages of the process, and the second a function of the terminal value of the system state at the end of the K'th stage. The multistage nature of this process is illustrated in Figure 8-1.

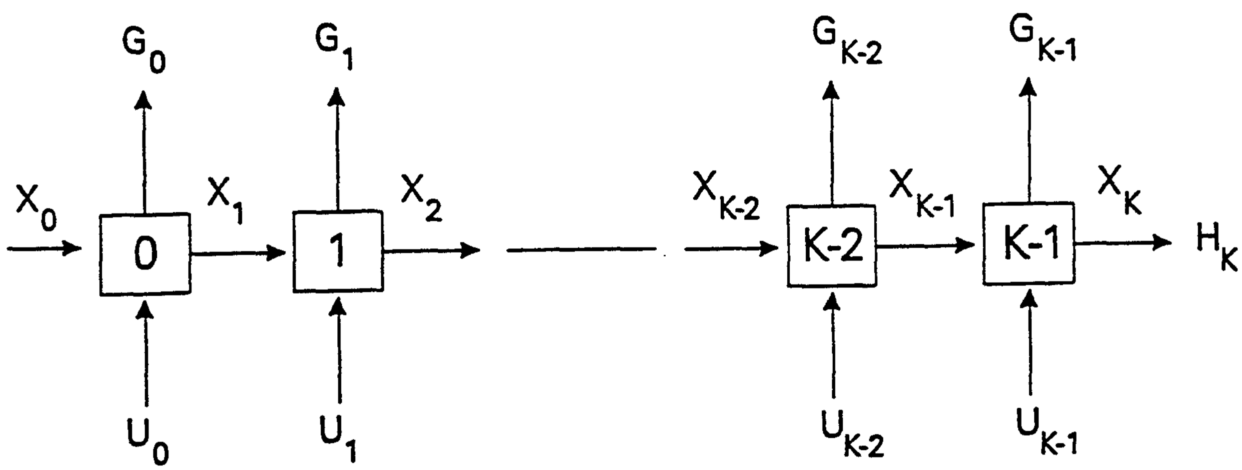


Figure 8-1. Multistage decision process.

In Figure 8-1 each stage of the process is represented by a block with an initial state x_i and a control input u_i. The output of each block consists of two parts, the next state x_{i+1}, obtained by means of the state transition equation, and a cost g_i which may depend on the initial state, the next state, and the control input. The terminal state of the process is x_K, and the cost associated with the terminal state is $h(x_k)$. By constructing different objective functions, enforcing different initial and final state conditions, and altering the number of stages over which the system is allowed to operate, a wide variety of optimization problems can be posed and in many cases solved by means of a dynamic programming algorithm.

The computational method of dynamic programming is based on Bellman's[8.5] principle of optimality which states that no matter how one arrived at a particular state, all decisions made from that state on must be made in an optimal manner. To compute the solution to an optimal control problem by means of dynamic programming, one begins at the end of the sequence, and for each possible state, determines the best control input which if applied minimizes the performance measure over the last stage and satisfies any required terminal conditions. The optimal solution for the last stage and each state are saved in a tabular format.

As a next step, the optimal solution is determined for the next to the last stage. This can be done because the performance measure increment attributable to the next to the last stage can be computed given each state and control variable combination, and the optimal control action and performance contribution to be used at the start of the last stage has been saved in tabular form.

The dynamic programming procedure repeats the computations and optimization done at the next to the last stage at each of the preceding stages back to the first stage. After all stage computations have been completed, the solution is obtained in the form of a table of control inputs to be applied when the dynamic system is in state x(k) at stage k.

As an example of applying the dynamic programming procedure, we will compute an approximate solution to the brachistochrone problem, illustrated in Figure 8-2. In this problem a friction-less bead having a constant mass slides down a wire from an initial starting point (x_a, y_a) to a lower ending point (x_b, y_b). The starting and ending points are fixed and the mass of the bead, m, and the acceleration of gravity, g, are assumed to remain constant. The problem is to determine the path or trajectory which results in the minimum travel time between the two specified points and the value of the minimum time.

In this example the initial point is located at x = 0 and y = 0 meters. The final point is at x = 2.5 and y = 1 meters. The mass of the bead is 1.0 kilogram and the acceleration of gravity is 9.8 meters/sec^2. Ten equal increments in x and one hundred equal increments in y were used to construct a grid of 1,111 points which covered the state space of interest. A stage of the decision process will correspond to the motion of the bead over one increment of x distance, 0.1 meters. This

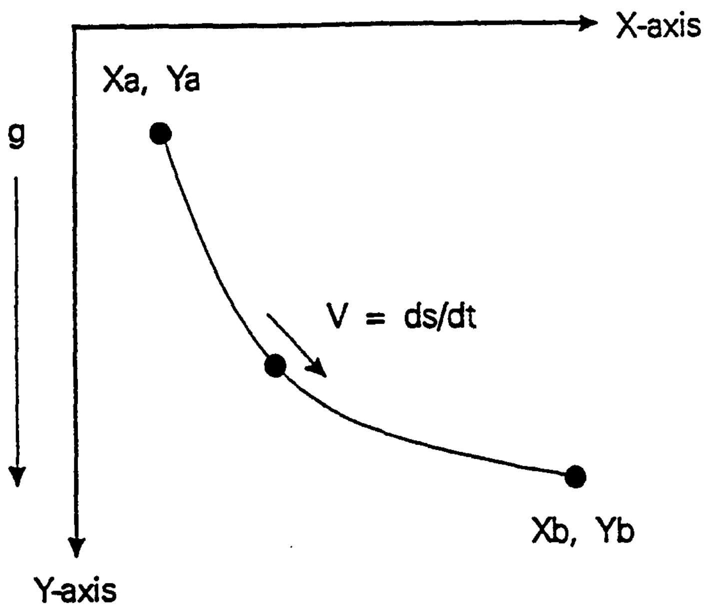


Figure 8-2. The brachistochrone problem.

is a ten-stage decision process. The state of the dynamic system is the y coordinate corresponding to the location of the bead at each position x_i.

The bead will be assumed to move with a varying velocity V between the coordinate pairs (x_1, y_1) and (x_2, y_2) at each stage of the process. The time to travel over any segment of the path is given by:

$$T = \left(\frac{1}{V}\right) \int_{x=x_1}^{x=x_2} ds \ ,$$

where ds is an incremental path length. Conservation of energy is assumed so that an increase in kinetic energy exactly equals a decrease in potential energy as the bead travels down the wire:

$$m\,g\,(y_2 - y_1) = \frac{1}{2}\,m\,V_2^3 - \frac{1}{2}\,m\,V_1^2, \text{ or}$$

$$V_2 = \left(2\,g\,(y_2 - y_1) + V_1\right)^{1/2} .$$

The velocity of the bead at any level y can then be written as V(y), and the initial velocity of the bead at y = 0 can be written as V_0. This gives a simpler expression for the velocity of the bead as a function of the coordinate y which applies to any path taken:

$$V(y) = (2\,g\,y + V_0)^{1/2} .$$

The increment of path length ds can be written in terms of the slope of the path, the derivative dy/dx:

$$ds = \left[1 + \left(\frac{dy}{dx}\right)^2\right]^{1/2} dx \ .$$

Combining all of the above gives another expression for the travel time over a segment of the path:

$$T = \int_{x=x_1}^{x=x_2} \frac{\left[1 + \left(\frac{dy}{dx}\right)^2\right]^{1/2}}{(2gy + V_0)^{1/2}} \, dx \ .$$

Straight line motion will be allowed between any pair of coordinates, and the equation of the line over which that motion takes place can be written as:

$$y = ax + b \ ,$$

where

$$a = \frac{(y_2 - y_1)}{(x_2 - x_1)} \text{ and } b = \frac{(y_1 x_2 - y_2 x_1)}{(x_2 - x_1)} \ .$$

Substituting this expression for y into the integral for T and evaluating the integral by means of a mathematical table look-up, yields an expression for the travel time between any pair (x_1, y_1) and (x_2, y_2) of initial and final coordinates:

$$T = \frac{(1 + a^2)^{1/2}}{g\,m} \left[\left(2gb + V_0^2 + 2gmx_2\right)^{1/2} - \left(2gb + V_0^2 + 2gmx_1\right)^{1/2} \right] .$$

The dynamic programming method can now be applied to yield an approximate solution to the brachistochrone problem. At the final stage, stage ten, the state of the system is constrained to be the pair of coordinates (x_{10}, y_{10}) or (2.5, 1.0). At the next to the last stage, stage nine, the state may take on any of the 101 coordinates given by (x_9, y_j) or $(2.25, y_j)$. The y_j represent all of the 101 grid values established for y at each stage. Applying the principle of optimality, the optimal trajectory segment for any initial coordinate at stage nine is a segment which steers the bead to the required end point, and the travel time for that segment is computed using the expression for T above. The travel time for each segment is saved for each grid point at stage nine, and that value represents the optimal solution for a minimum time trajectory which begins in stage nine.

The method is then applied at stage eight in a similar manner. For each possible starting point at stage eight and each possible ending point (at stage nine), the total travel time to reach the specified end point is computed. This value is the sum of two terms, the time over the segment from stage eight to stage nine plus the previously stored travel time from stage nine to stage ten. The minimum total travel time and the optimal next state at stage nine are saved for each grid point at stage eight. This process is repeated in a recursive manner for all the remaining stages from seven back to zero. Note that at stage zero only the specified starting point needs to be considered, and an examination of the travel times reveals that no segments which travel upward through the state space need be evaluated.

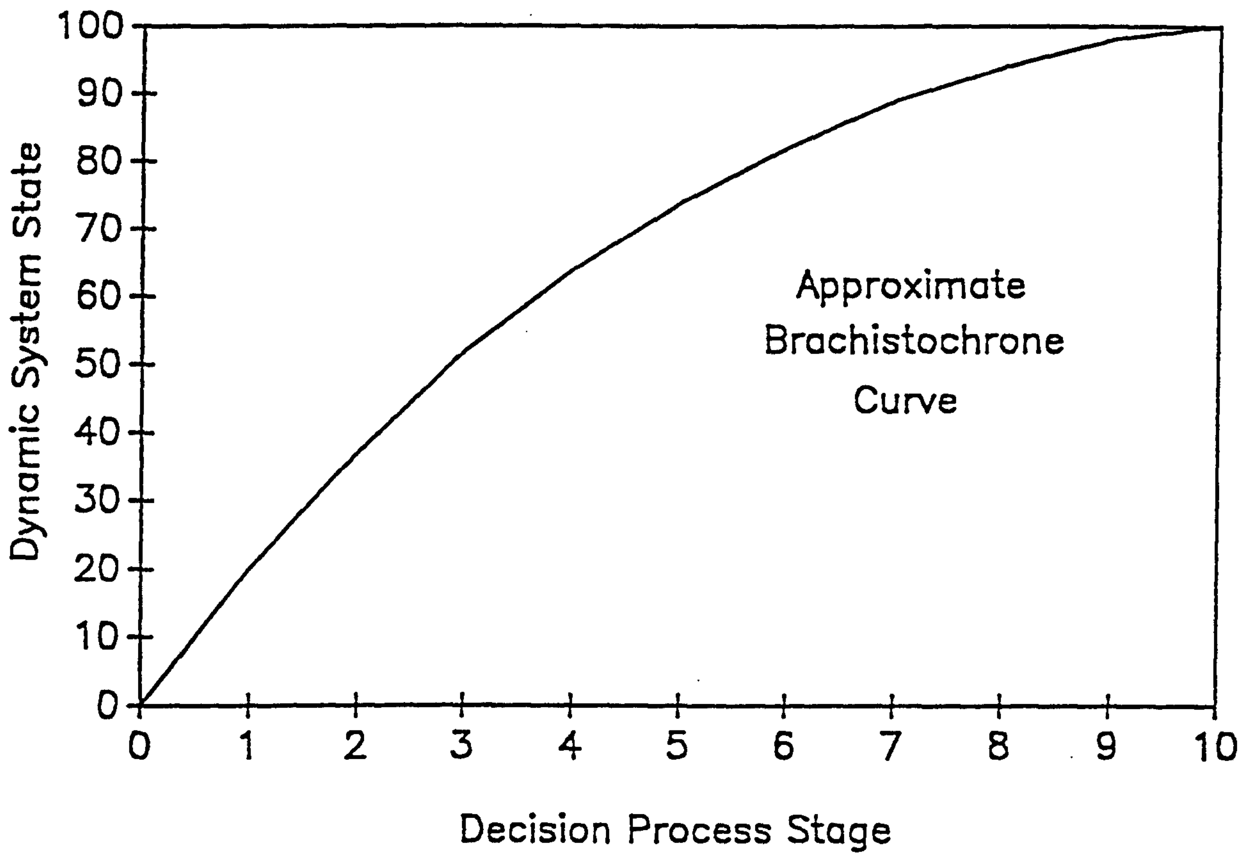


Figure 8-3. Dynamic programming solution of system states versus decision stages.

The dynamic programming solution to the brachistochrone problem obtained by this process is illustrated in Figures 8-3 and 8-4. The first figure plots the dynamic system state in terms of the integer grid point index j versus the stage of the decision process. The second figure plots this same solution in terms of the x and y coordinates. In this figure the y-axis is plotted upwards rather than downwards as in the illustration of Figure 8-2. Note that the trajectory obtained is relatively smooth, a consequence of taking a fine grid of 101 values for the y-coordinate. The initial velocity V_0 was taken to be 5.0 meters per second.

The dynamic programming method also produces the value of the performance measure for each potential grid point. Figure 8-5 illustrates the manner in which the performance measure, minimum travel time, varies as a function of the decision process stage along the optimal trajectory.

The data recorded for each grid point, the next state, and the value of the performance measure provide a table look-up closed-loop controller for the dynamic system represented by the moving bead. The solution obtained includes the optimal trajectories for all grid points considered. If the bead were initially placed in a different starting state, the optimal trajectory and minimum time solution for that starting state is also available. This aspect of the dynamic programming procedure is highly valuable if alternate trajectories are of interest.

8.7 Summary

Mathematical optimization is so-named because it is a branch of modern control theory that makes use of specific mathematical procedures such as minimization, maximization, linear programming, nonlinear programming, method of least squares, method of steepest descent, calculus of variations, Bellman's method, and numerical integration. These procedures were applied to a variety of problems in this chapter. Minimization was used to solve a static, non-time-dependent problem. Linear programming was applied to a dynamic control problem for curve tracking. Calculus of variations was applied to a trajectory problem with defined endpoints using the Euler-Lagrange equation. Another example concerned the use of calculus of variations in dynamic programming of optimal control laws. Dynamic programming was also used to solve a brachistochrone problem.

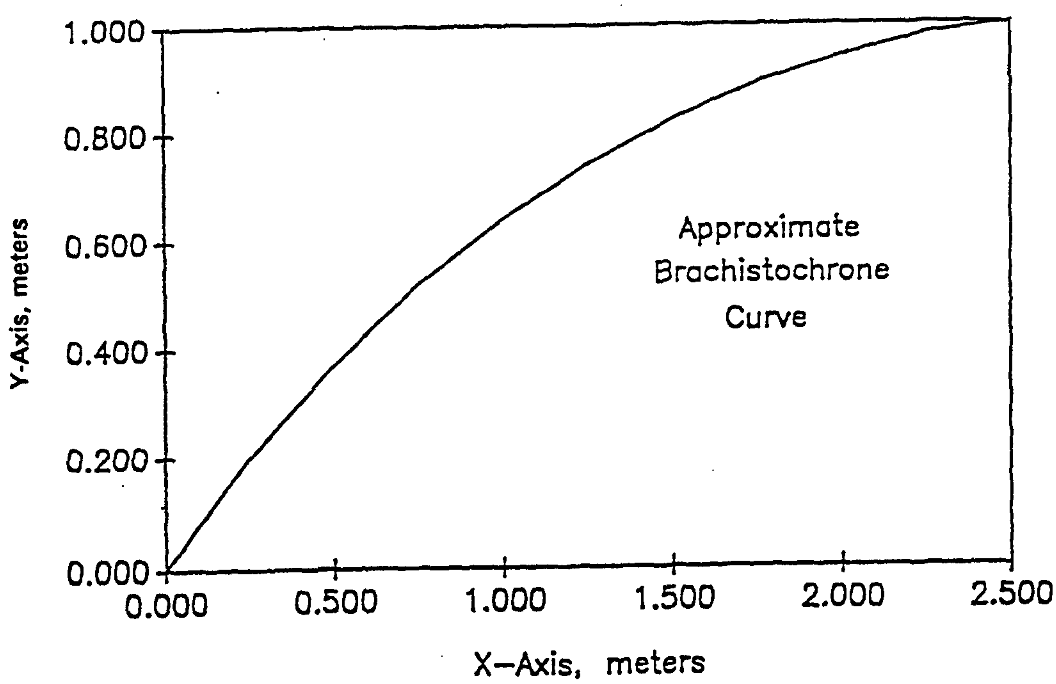


Figure 8-4. Dynamic programming solution in x-y coordinates.

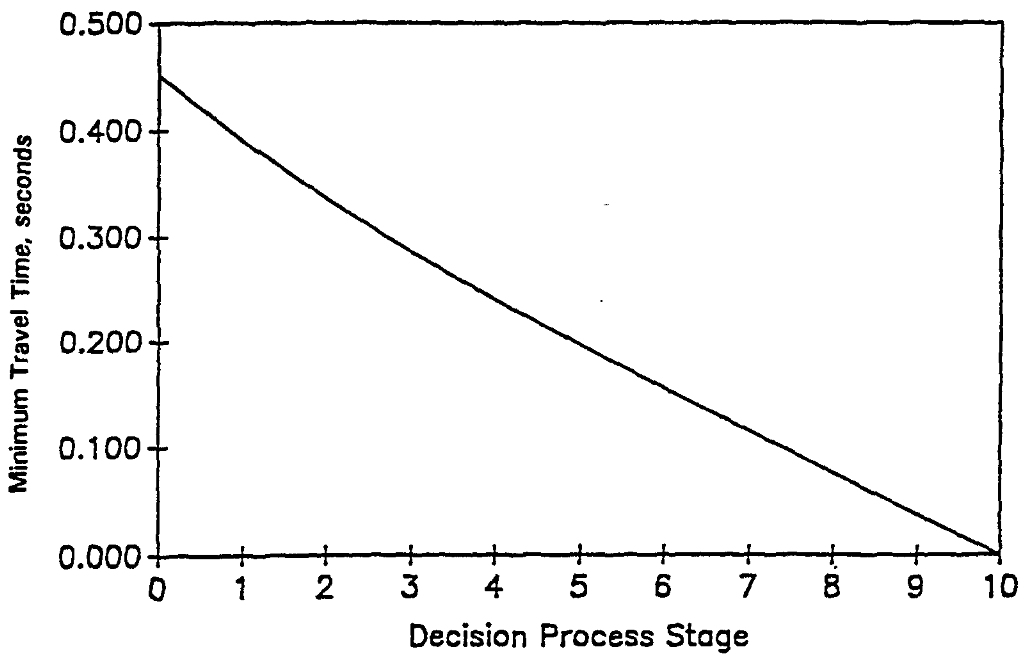


Figure 8-5. Dynamic programming solution for minimum travel time for a brachistochrone problem.

REFERENCES

8.1 Kailath, T., Linear Systems, Prentice-Hall, Englewood Cliffs, NJ, 1980.

8.2 Chen, C.T.., Linear System Theory and Design, Holt, Rinehart and Winston, 1984.

8.3 Kirk, D.E., Optimal Control Theory, An Introduction, Prentice Hall, Inc., Englewood Cliffs, New Jersey, 1970.

8.4 Koo, D., Elements of Optimization, With Applications to Economics and Business, Springer-Verlag, New York, 1977.

8.5 Bellman, R.E., Glicksburg, I., and Gross, O.A., On the "Bang-Bang" Control Problem, Quarterly of Applied Mathematics, Vol.14, pp.11-18, 1956.

This page is intentionally blank.

CHAPTER 9
OPTIMAL CONTROL THEORY

9.1 Introduction

The process of mathematical modeling and state variable analysis yields a mathematical model of a dynamic system in the form of a set of state transition equations. For a continuous-time process, these state transition equations will consist of first-order differential equations. For a discrete-time process, a set of first-order difference equations will be produced. For example, the simplest linear differential equation has the form:

$$\frac{dx(t)}{dt} = -a\,x(t), \; x(0) = x_0 \; .$$

This equation describes a system having a single state variable x, whose initial value is x_0. The solution to this differential equation is the familiar exponential decay process when the parameter a is greater than zero:

$$x(t) = x_0 e^{-at}, \; t \geq 0 \; .$$

For different initial conditions, the response of the dynamic system will follow a different trajectory, decaying eventually to the state $x = 0$.

To obtain control over this dynamic system it is necessary to introduce a control input into the model. The differential equation describing this modified process is:

$$\frac{dx(t)}{dt} = -a\,x(t) + b\,u(t), \; x(0) = x_0 \; .$$

The solution to this differential equation has the form:

$$x(t) = x_0 e^{-at} + \int_{\tau=0}^{\tau=t} b\, e^{-a(t-\tau)}\, e(\tau)\, d\tau \; .$$

The first term in this expression represents the response of this system when the system state is initially x_0 and no input is introduced. This response is referred to as the zero-input or natural response. The second term represents the response due to the application of a control input. This response is referred to as the zero-state or forced response. If a control input u(t) is selected and applied when the system starts in a state x_0, the response will have a predictable form. If the same control input is applied when the system is initially in a different state the response to the same control input will, as a whole, be different.

Optimal control involves the selection of a particular control input for a dynamic system such as the one above. This selection is made so as to optimize a specified performance measure which can be a function of the state trajectory, the applied control input, the final system state, and the time required to reach that state. The particular performance measure used is selected by the control system designer, or provided as part of the problem statement.

The optimal control input may be determined as an open-loop control action generated and applied strictly as a function of time, or as a closed-loop control action generated and applied as a function of the dynamic system state. Closed-loop controls are also called feedback controls. Classical control system design methods are primarily concerned only with the design of feedback controls for linear time-invariant dynamic systems. Classical design methods cannot be directly applied to the design of controllers for nonlinear, time-varying dynamic systems. For these systems, optimal control often provides the only useful design approach.

Constraints will often be present which limit the available set of control actions which may be considered. For example, the control action may be limited in magnitude and, after scaling, be required to lie between an upper and a lower value of 1. This constraint can be written as:

$$|u(t)| \leq 1.0, \; t \geq 0 \; .$$

The state of the dynamic system may also be constrained to lie within some limits, for example:

$$|x(t)| \leq 5.0, \; t \geq 0 \; .$$

Such constraints reflect limitations posed by operational considerations or hardware design.

9.2 Development of Optimal Control Theory

Optimal control theory traditionally involves the control of deterministic dynamic systems whose evolutions are modeled by ordinary differential equations. The results and methods of traditional optimal control theory have gradually been extended to encompass deterministic discrete-time dynamic systems modeled by sets of difference equations and problems of stochastic control in which the state transition process is random, either as the result of a noise process or due to random transitions from state to state. This chapter of this review mainly considers deterministic dynamic systems having the following state transition structure:

$$\frac{dx(t)}{dt} = f\big(x(t),\ u(t),\ t\big),\ x(0) = x_0 \ .$$

A control input u(t) is to be selected so as to minimize a performance measure, the functional:

$$J = h\big(x(T),\ T\big) + \int_{\tau=0}^{\tau=T} g\big(x(\tau),\ u(\tau),\ \tau\big)\, d\tau \ .$$

Various specific forms of this functional will be exhibited later when specific problems in optimal control are discussed.

The initial development and application of optimal control theory occurred from about 1950 to 1962. Initial efforts focused on improving the response of servomechanisms. The state transition equation for these early problems can be written as a linear, time-varying differential equation:

$$\frac{dx(t)}{dt} = A(t)\,x(t) + B\,u(t) \ ,$$

with $x(0) = x_0$

and $|u(t)| \le C$.

In this problem the state x(t) is represented by a vector of n components and the control input u(t) by a vector of m components. The system is an m-input, n-output multivariable system. The initial state of the system is x_0 and a final or terminal state x(T) is often specified. The final time T is known in advance, and C is a vector of limits which constrain the control input magnitudes.

If the functional J is selected such that $h = 0$ and $g = 1$, the performance measure becomes:

$$J = \int_{\tau=0}^{\tau=T} 1\, d\tau \ ,$$

and the problem becomes one of minimizing the time necessary to drive the system from the initial state x_0 to the final state x(T) when the available control inputs are limited. This particular problem is called the time-optimal control problem.

In general, an optimal solution to the time-optimal problem may not exist. From a practical point of view the limited control inputs may simply be insufficient to produce the required change in the system's state, or the specified final state may be unreachable in the time demanded.

Early on, much effort was devoted to determining conditions under which the existence of a solution was assured, and ways in which to compute and implement the optimal control policy. Bellman[9.1] studied a version of this time-optimal problem in which the matrices A and B were constant, the eigenvalues of A all had negative real parts, B was a square, non-singular matrix, and the control inputs were constrained by C = 1.

Bellmann showed that for this problem an optimal solution existed and was a bang-bang control law, in which each component of u took on a value of either +1 or −1 at each point in the control interval. Bellman developed a mathematical technique for computing the times at which switchings between the limiting values occurred. Bang-bang control laws have since been proven to be the form of the optimal control law for the minimum time control of a wide class of linear dynamic systems.

Gamkrelidze[9.2] of the USSR studied the time-optimal problem for linear, continuous-time, time-invariant systems in which A and B were constant matrices, without the requirement of a non-singular B matrix. He also derived a bang-bang optimal control law. Krasovskii[9.3] considered the time-optimal control problem for linear, continuous-time, time-varying dynamic systems with the further requirement that it was necessary to hit a moving target whose position or state was given by z(t).

LaSalle[9.4] generalized all previous results concerning the time-optimal control problem. He showed that if a moving target can be hit at all, it can be hit in minimum time by a control input generated according to a bang-bang control law. His specific bang-bang control law requires the components of u to be either +1 or −1 almost everywhere.

Filippov[9.5] established the existence of time-optimal control laws for nonlinear dynamic systems. He also introduced methods and ideas used to investigate the existence of optimal control laws for more general problems.

9.3 Pontryagin's Maximum Principle

The Soviet mathematicians Pontryagin, Boltyanski, Gamkrelidze, and Mischenko published a series of technical papers during the period from 1956 to 1960 in which they investigated the general nonlinear optimal control problem and developed what was to become known as Pontryagin's maximum principle[9.6]. This principle is a set of necessary mathematical conditions that must be satisfied by a control input and state variable trajectory, or time history, if that control input is to minimize the problem's performance measure. Their work involves the introduction of a set of costate variables which serve the same purpose as Lagrange multipliers in a static optimization problem, and an auxiliary function, H, called the Hamiltonian.

The maximum principle approach allows the necessary conditions which must be satisfied by the solution to a general optimal control problem to be written in a compact form. This procedure will be demonstrated by means of an example.

Let the dynamic system be represented by the following state transition equations:

$$\frac{dx(t)}{dt} = f\big(x(t),\ u(t),\ t\big),\ x(0) = x_0 .$$

Let the performance measure be written as:

$$J = h\big(x(T),\ T\big) + \int_{\tau=0}^{\tau=T} f\big(x(\tau),\ u(\tau),\ \tau\big)\, d\tau .$$

In this general optimal control problem the state vector x is n-dimensional and the control input vector u is m-dimensional. Define an auxiliary n-dimensional vector, the costate vector, as p(t) and introduce the Hamiltonian function H defined by:

$$H\big(x(t),\ u(t),\ p(t),\ t\big) = \frac{\partial J}{\partial \tau} + p^T(t)\,\frac{dx(t)}{dt},\ \text{or}$$

$$H\big(x(t),\ u(t),\ p(t),\ t\big) = g\big(x(t),\ u(t),\ t\big) + p^T(t)\, f\big(x(t),\ u(t),\ t\big) .$$

By taking a derivative with respect to time it can be shown that the costate vector satisfies the following differential equation:

$$\frac{dp(t)}{dt} = -\frac{dH\big(x(t),\ u(t),\ p(t),\ t\big)}{dt}$$

$$= -\frac{\partial g\big(x(t),\ u(t),\ t\big)}{\partial x(t)} - p^T(t)\,\frac{\partial f\big(x(t),\ u(t),\ t\big)}{\partial x(t)} .$$

Pontryagin's maximum principle states that if an admissible control input $u^*(t)$ and its resulting state variable trajectory $x^*(t)$ are optimal with respect to maximizing a performance measure J, then there also exists a non-zero costate vector $p^*(t)$ corresponding to $u^*(t)$ and $x^*(t)$ such that for all times t from t = 0 to t = T:

a) $\dfrac{dx(t)}{dt} = \dfrac{\partial H}{\partial p}$,

b) $\dfrac{dp(t)}{dt} = -\dfrac{\partial h}{\partial x}$,

c) The function H attains its maximum when evaluated along $u^*(t)$ compared to any other control input u(t),

d) $\dfrac{\partial h\left(x*(t_f),\ t_f\right)}{\partial x} = p*(t_f)$, and

e) $x(0) = x_0$ as specified.

Pontryagin's maximum principle allows these necessary conditions to be written quickly and in a highly compact manner. Unfortunately, the maximum principle only yields a set of necessary conditions which must be satisfied by the optimal solution. The maximum principle provides no guidance as to how the solution of the optimal control problem is to be obtained. For virtually all practical problems of interest, the development and application of sophisticated numerical methods is required to arrive at an approximate solution to the optimization problem.

Pontryagin's maximum principle has been applied to solve the linear time-optimal control problem. This optimal control problem was one of the first optimal control problems to be studied in detail. For a dynamic system having one or two state variables, the solution obtained via Pontryagin's maximum principle can be illustrated graphically. For problems involving higher-dimensional state spaces, the solution can only be computed numerically. The solution presented below illustrates the use of Pontryagin's maximum principle to establish a set of necessary conditions and the use of these conditions to solve for the form of the optimal control input.

For the time-optimal control problem the performance measure to be minimized is:

$$J = \int_{\tau=0}^{\tau=t_f} 1\, d\tau = -\int_{\tau=0}^{\tau=t_f} (-1)\, d\tau\ .$$

The final time t_f is free and to be minimized. Since minimization of a performance measure is equivalent to maximization of the negative of that same performance measure, use:

$$J = -\int_{\tau=0}^{\tau=t_f} 1\, d\tau = -t_f\ .$$

The dynamic system to be controlled is a second-order time-invariant linear system having the following set of state transition equations:

$$\frac{dx_1(t)}{dt} = x_2(t),\ x_1(0) = x_{10}\ ,$$

$$\frac{dx_2(t)}{dt} = u(t),\ x_2(0) = x_{20}\ ,$$

and the control input is limited by the constraint:

$$|u(t)| \leq 1 .$$

The initial state $x(0)$ is unspecified, but assumed to be given, and the required final state is $x(t_f) = 0$. Since there are two state variables, two costates are introduced, $p_1(t)$ and $p_2(t)$ and the Hamiltonian is constructed:

$$H = -1 + p_1(t)\frac{dx_1(t)}{dt} + p_2(t)\frac{dx_2(t)}{dt} \text{ , or}$$

$$H = -1 + -p_1(t)\,x_2(t) + p_2(t)\,u(t) .$$

The costates satisfy the following differential equations:

$$\frac{dp_1(t)}{dt} = -\frac{\partial H}{\partial x_1} = 0, \text{ and}$$

$$\frac{dp_2(t)}{dt} = -\frac{\partial H}{\partial x_2} = -p_1(t) .$$

The solution to the costate differential equations is:

$$p_1(t) = \text{constant} = c_1, \text{ and}$$

$$p_2(t) = -c_1\,t + c_2 .$$

The optimal control action $u^*(t)$ maximizes the Hamiltonian. Since $u(t)$ is multiplied by $p_2(t)$ in the Hamiltonian, H will be maximized if the sign of $u^*(t)$ is opposite that of $p_2(t)$, and if $u^*(t)$ takes on its largest possible magnitude. This result can be written mathematically as:

$$u^*(t) = \begin{bmatrix} +1, & p_2(t) \geq 0, \\ -1, & p_2(t) < 0 \end{bmatrix} .$$

This expression for $u^*(t)$ is valid for all times t in the control interval from $t = 0$ to $t = t_f$.

Since $u^*(t)$ is either $+1$ or -1 it is easy to solve the state transition equations. When $u^*(t)$ equals $+1$:

$$\frac{dx_2^*(t)}{dt} = +1, \text{ or } x_2^*(t) = t + k_2, \text{ and}$$

$$\frac{dx_1^*(t)}{dt} = x_2^*(t), \text{ or}$$

$$x_1^*(t) = \frac{t^2}{2} + k_2 t + k_1 = \frac{1}{2}(t+k_2)^2 + \left[k_1 - \frac{k_2^2}{2}\right], \text{ or}$$

$$x_1^*(t) = \frac{1}{2}\, x_2^*(t)^2 + k_3 \,.$$

When u*(t) equals −,

$$\frac{dx_1^*(t)}{dt} = -1, \text{ or } x_2^*(t) = -t + k_4, \text{ and}$$

$$\frac{dx_1^*(t)}{dt} = x_2^*(t), \text{ or}$$

$$x_1^*(t) = -\frac{1}{2}\, x_2^*(t)^2 + k_5 \,.$$

The coupled motions of $x_1^*(t)$ and $x_2^*(t)$ for $u^*(t) = +1$ and $u^*(t) = -1$ form two families of parabolas as shown in Figure 9-1[9.7].

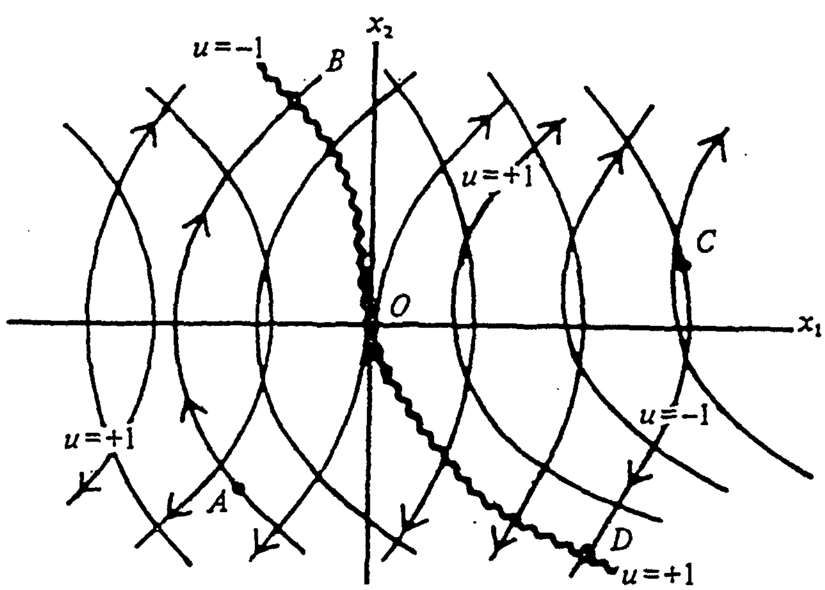


Figure 9-1. Parabolic Pontryagin solutions to a bang-bang control strategy.

The state $x^*(t) = [x_1^*(t), x_2^*(t)]^T$ moves upwards in the direction indicated by the arrows when $u^*(t)$ equals +1 and downwards when $u^*(t)$ equals −1. The minimum time path from any arbitrary starting state to the origin is obtained graphically by moving along one arc of a parabola passing through the starting state until a parabola passing through the origin is reached. At that point in the state space, the control input switches sign and the state follows the arc of the second parabola to the origin. This control strategy, switching between the maximum values of the control input, is referred to as a bang-bang control strategy.

Note that the solution has yielded the optimal state trajectory and control input but not the value of the minimum time. The bang-bang control strategy can be implemented as a closed-loop control policy since the switching points depend on the state of the dynamic system, rather than on time as a parameter.

The form of the optimal control input can be written as function of the two state variables in feedback form:

$$u(x_1, u_2) = \begin{bmatrix} +1, \text{ if } x_2 > 0,\ x_1 < \frac{1}{2}x_2^2 \\ +1, \text{ if } x_2 < 0,\ x_1 \leq \frac{1}{2}x_2^2 \\ -1, \text{ otherwise} \end{bmatrix}, \text{ and}$$

a switching function $S(x_1, x_2)$ can be defined by:

$$S(x_1, x_2) = x_1 + \frac{1}{2}x_2 \operatorname{abs}(x_2)\ .$$

The role of this switching function can be seen in the diagram of the time-optimal controller in Figure 9-2.

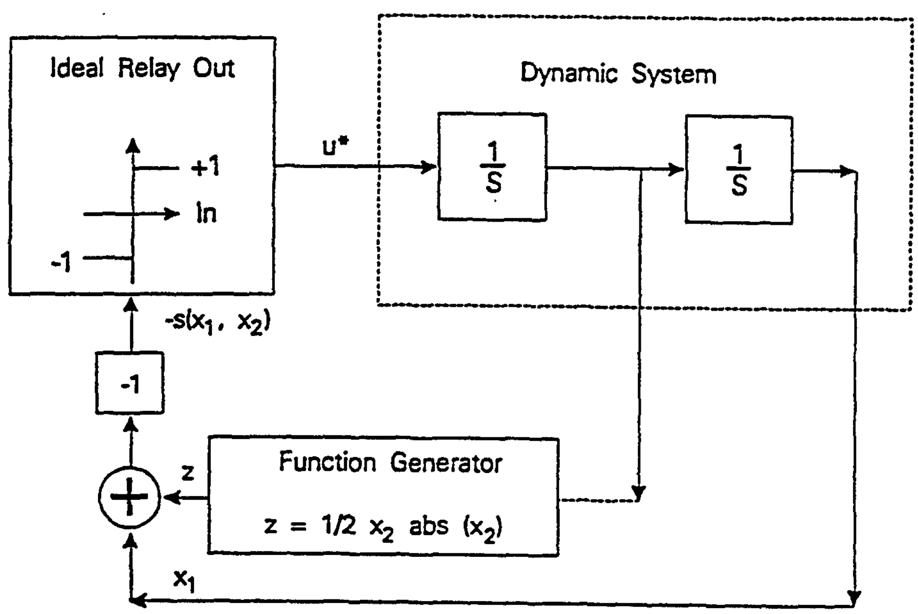


Figure 9-2. Structure of the time-optimal controller.

Figure 9-3 illustrates the shape of this switching function. The switching function can be used to write the optimal control policy as:

$$u^*(t) = \begin{bmatrix} -1, \text{ if } S(x_1, x_2) > 0, \\ \quad \text{or } S(x_1, x_2) = 0, \text{ and } x_2 > 0, \\ +1, \text{ if } S(x_1, x_2) < 0 \\ \quad \text{or } S(x_1, x_2) = 0 \text{ and } x_2 < 0, \\ 0, \text{ if } x_1 = 0 \text{ and } x_2 = 0 \text{ .} \end{bmatrix}$$

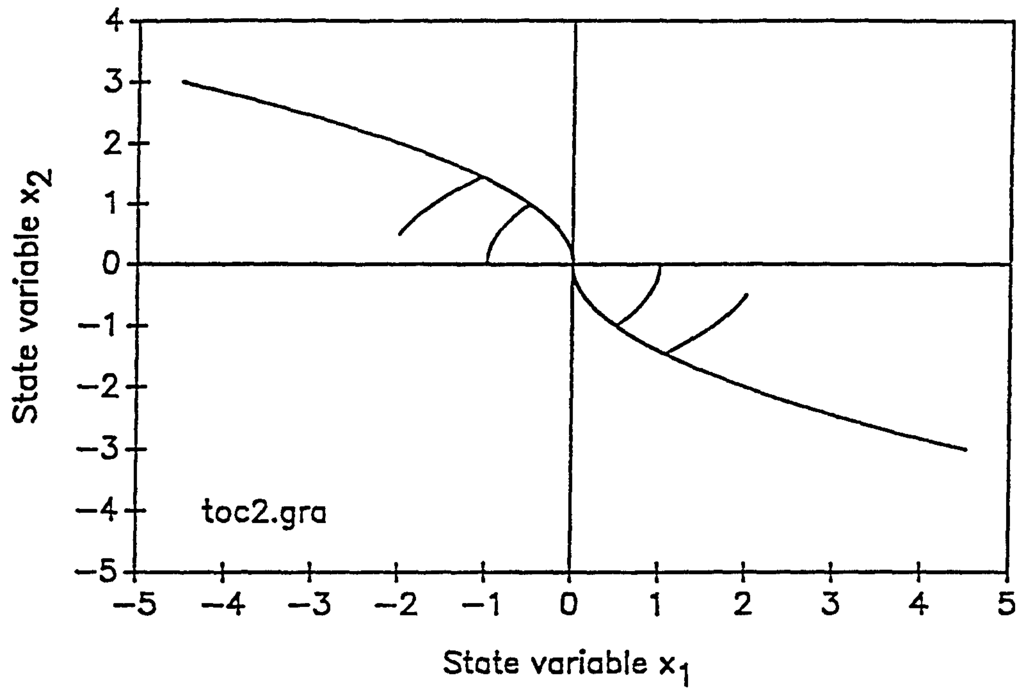


Figure 9-3. Switching functions for time-optimal state tranjectories.

Figure 9-4 shows an implementation of the optimal controller for this process. The controller evaluates the switching function and determines the optimal control input as a function of the system state variables. Note that both state variables must be measured to implement this control policy. Additional examples of linear time-optimal control problems and their solution can be found in Kirk[9.8].

As a second example[9.8 page 234] of the application of Pontryagin's maximum principle, consider the optimal control of a second-order linear time-invariant dynamic system having the following state transition equations:

$$\frac{dx_1(t)}{dt} = x_2(t),\; x_1(0) = x_{10} \text{ ,}$$

$$\frac{dx_2(t)}{dt} = -x_2(t) + u(t), \; x_2(0) = x_{20} \; ,$$

The performance measure to be minimized is:

$$J = \int_{\tau=0}^{\tau=t_f} \frac{1}{2}\left[x_1^2(\tau) + u^2(\tau)\right] d\tau \; .$$

The final time t_f is assumed to be specified and the final state $x(t_f) = [x_1(t_f), x_2(t_f)]^T$ is free. The control objective indicated by the performance measure is to drive the dynamic system state x(t) as close to the origin of the state space as possible while minimizing the control energy exerted. The performance measure is the sum of two equally weighted terms reflecting the importance of each control objective.

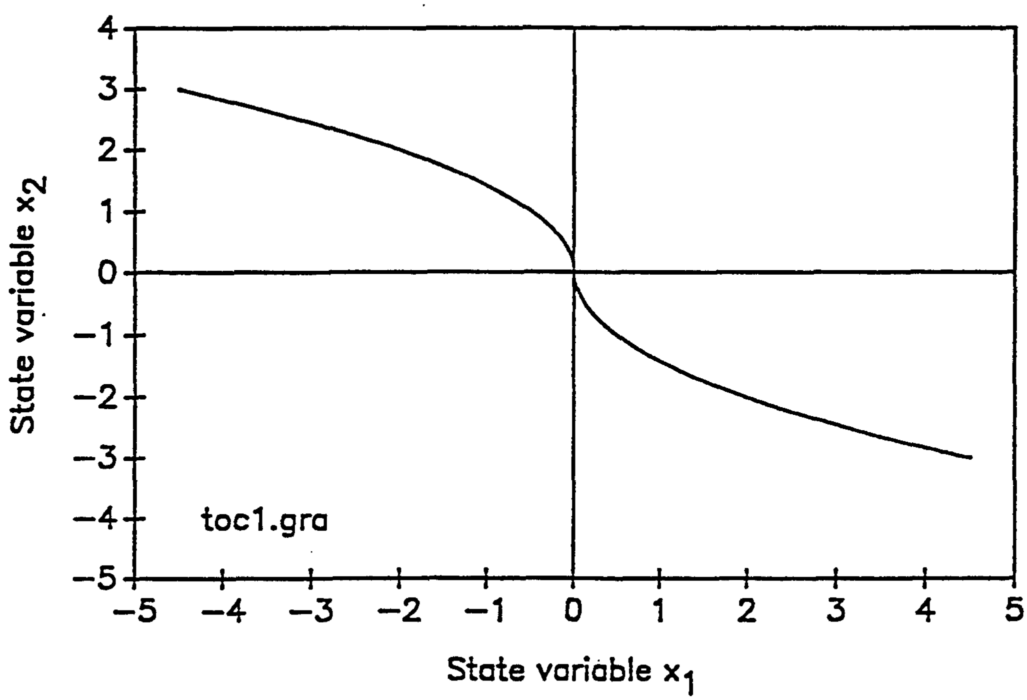


Figure 9-4. Time-optimal switching function.

Pontryagin's maximum principle can be applied to write a compact set of necessary conditions which must be satisfied by the optimal solution to this problem. Introduce two costates and write the Hamiltonian function:

$$H = -\frac{1}{2}\left[x_1^2(t) + u^2(t)\right] + p_1(t)\,x_2(t) + p_2(t)\left[-x_2(t) + u(t)\right] .$$

The necessary conditions for optimality are:

a) $\frac{dx}{dt} = -\frac{\partial H}{\partial p}$, or

$$\frac{dx_1(t)}{dt} = x_2(t),$$

$$\frac{dx_2(t)}{dt} = -x_2(t) + u(t),$$

b) $\frac{dp}{dt} = -\frac{\partial H}{\partial x}$, or

$$\frac{dp_1(t)}{dt} = -x_1(t),$$

$$\frac{dp_2(t)}{dt} = -p_1(t) + p_2(T),$$

c) $\frac{\partial H}{\partial u} = 0 = u(t) + p_2(t)$,

since the control input u(t) is unconstrained, or

$$u(t) = -p_2(t),$$

d) $\frac{\partial h\left(x^*(t_f), t_f\right)}{\partial x} = p^*(t_f) = 0$, and

e) $x(0) = x_0$ as specified.

The solution to this optimal control problem requires the solution to a linear two-point boundary value problem. The control input u(t) is completely determined by the costate $p_2(t)$ and u(t) can thus be eliminated from the equation during the solution process. The boundary conditions are split since x(0) is specified at the time when t equals zero and $p(t_f)$ equals zero at the final time t_f.

If the control input is constrained by:

$$|u(t)| \leq 1, \; 0 \leq t \leq t_f,$$

the state and costate equations, the Hamiltonian function, and the boundary conditions for this problem remain the same as those indicated above. The main effect of the constraint is that it is no longer always possible to take the partial derivative of the Hamiltonian with respect to u(t). Pontryagin's maximum principle can still be invoked and that u(t) which minimizes the Hamiltonian can be determined by investigating the structure of the Hamiltonian. The Hamiltonian for this problem is:

$$H = -\frac{1}{2}\left[x_1^2(t) + u^2(t)\right] + p_1(t)\,x_2(t) + p_2(t)\left[-x_2(t) + u(t)\right].$$

Collecting those terms which involve the control action u(t) we have:

$$-\frac{1}{2}\,u^2(t) + p_2(t)\,u(t).$$

When the optimal control action u(t) lies within the bounds imposed by the constraint, the Hamiltonian will be maximized if u(t) takes on the sign opposite to $p_2(t)$ and simultaneously assumes the magnitude of $p_2(t)$. If the magnitude of $p_2(t)$ is greater than one, the control input u(t) must be limited. Combining all of this, an expression for the optimal control policy can be written:

$$u^*(t) = \begin{bmatrix} +1,\ p_2^*(t) < -1, \\ -p_2^*(t),\ -1 < p_2^*(t) < +1, \\ -1,\ +1 < p_2^*(t) \end{bmatrix}.$$

A two-point boundary value problem must again be solved to obtain a numerical solution to this optimal control problem. Since the control is constrained, it cannot be simply eliminated from the state transition equations. The resulting nonlinear two-point boundary value problem is difficult to solve, and the solution cannot be obtained in general by solving the unconstrained control problem and then passing the resulting optimal control solution through a limiting process.

9.4 Open-Loop and Closed-Loop Optimal Control

The solution to the general optimal control problem for a deterministic system consists of a control input, $u^*(t)$, which is found based on a specified initial state of the dynamic system, x_0, and is generated and applied as a known function of time once computed at the start of the problem. In many applications, it is impossible to accurately know the initial state of the dynamic system precisely, due to measurement inaccuracies or random disturbances not accounted for in the computation of the optimal control. For these reasons, it is useful to design the control system so that the optimal control input is computed as a function of the state of the system at time t, or x(t).

To illustrate this point, we consider a general optimal control problem having the following form:

$$\frac{dx(t)}{dt} = f\big(x(t),\, u(t)\big),\ x(0) = x_0,$$

with a performance measure of the form:

$$J = \int_{\tau=0}^{\tau=T} g\big(x(\tau),\ u(\tau)\big)\, d\tau\ .$$

An open-loop optimal control is then understood to be a piece-wise continuous function of time defined over the time of control which drives the dynamic system starting in the state x_0 and minimizes the performance measure J. This implies that no other control input can do any better in terms of minimizing J.

Unfortunately, it is very difficult to determine optimal controls in feedback form. Two special control problems, the linear quadratic optimal control problem and the linear time-optimal control problem, do have optimal control laws which specify the control input as a function of the present state of the system.

If an optimal control problem can be reduced to or approximated by one of these special problems, then it will be possible to derive a suboptimal feedback controller. For that reason, these two problems play an important role as the basis for many other optimal control formulations and the development and implementation of optimal control algorithms. The difficulty of determining generalized closed-loop feedback optimal solutions in many cases of interest is one reason why optimal control policies have not seen wider application.

9.5 Performance Measures and Optimal Control Problems

The general optimal control problem involves finding a control action u*(t) which causes the dynamic system defined by the state transition equation:

$$\frac{dx(t)}{dt} = f\big(x(t),\ u(t),\ t\big),\quad x(0) = x_0$$

to follow a state trajectory x*(t) which minimizes a performance measure of the form:

$$J = h\big(x(t_f),\ t_f\big) + \int_{\tau=0}^{\tau=t_f} g\big(x(\tau),\ u(\tau),\ \tau\big)\, d\tau\ .$$

In the application of optimal control, the state transition equations which define the dynamic system are treated as being given as part of the problem specification. The control system designer is usually not at liberty to extensively modify the underlying dynamic system structure. The designer may have some ability to select the performance measure for the problem. By selecting different performance measures, different optimal control problems can be developed. In the following subsections several performance measures commonly encountered in the optimal control literature are presented and related to optimal control problems of interest.

9.5.1 Minimum Time Problem

In the minimum time optimal control problem the objective is to transfer the state of the dynamic system from an arbitrary initial state x(0) to a specified terminal state $x(t_f)$ in the minimum amount of time. The performance measure to be minimized is:

$$J = \int_{\tau=0}^{\tau=t_f} 1 \, d\tau \, .$$

The final time t_f is not specified but is taken as the first instant at which the terminal state $x(t_f)$ is reached.

The minimum time problem has been used as the structure for optimal control problems involving the intercept of attacking aircraft and missiles and for problems involving the rapid response of servomechanisms associated with the slewing motions of radar antennas, missile launchers, and gun mounts.

9.5.2 Terminal Control Problem

The objective in a terminal control problem is to minimize the deviation of the terminal state $x(t_f)$ of a dynamic system from a desired terminal state $r(t_f)$. The final time t_f may be either infinite or finite. A commonly encountered performance measure to be minimized for a terminal control problem is:

$$J = \left[x(t_f) - r(t_f)\right]^T H \left[x(t_f) - r(t_f)\right] ,$$

where H is a real, symmetric, positive semi-definite n by n weighting matrix. A real, symmetric matrix H is called positive semi-definite if for all vectors z, the scalar (z^T H z) is greater than or equal to zero. For such a matrix, there are some vectors z for which the matrix product (H)z equals zero, and for all others the scalar (z^T H z) is greater than zero. When the matrix H has only diagonal elements the performance measure is a weighted sum-of-the-squares of the terminal state variables, and the weights reflect the relative importance if each component can serve as scaling factors so that all state variables are measured to a common reference.

Terminal control problems arise when attempting to model and control the launch of tactical guided weapons against stationary targets whose coordinates are known in advance.

9.5.3 Minimum Control Effort Problems

When the applied control input represents a control effort such as force, torque, thrust, or fuel consumption it is natural to use a performance measure based solely on the applied control and which leads to the conservation of a scarce resource. If there is a single control input which may take on both positive and negative values a performance measure similar to the following is appropriate for minimization:

$$J = \int_{\tau=0}^{\tau=t_f} |u(\tau)| \, d\tau .$$

In this performance measure the variable sign of $u(\tau)$ is accounted for by the absolute value function and both positive and negative control input excursions are equally weighted. When multiple control inputs are involved the performance measure to be minimized is written as:

$$J = \int_{\tau=0}^{\tau=t_f} \sum_{i=1}^{i=m} c_i \, |u_i(\tau)| \, d\tau .$$

The weights c_i are selected by the designer to reflect the relative importance of each control input u_i.

When the control input represents a voltage or current and the dynamic system is to be controlled in a manner which minimizes the total energy dissipation, the following performance measure is minimized:

$$J = \int_{\tau=0}^{\tau=t_f} u^2(\tau) \, d\tau .$$

When multiple control inputs are involved and it is desired to minimize the total control energy expended, the appropriate performance measure is:

$$J = \int_{\tau=0}^{\tau=t_f} u^T(\tau) \, R(\tau) \, u(\tau) \, d\tau ,$$

where $R(\tau)$ is a real, symmetric, positive definite, possibly time-varying weighting matrix.

9.5.4 Tracking and Regulator Problems

The objective in a tracking optimal control problem is to maintain the dynamic system state $x(t)$ as close as possible to a desired state $r(t)$ over the control interval from $t = 0$ to $t = t_f$. The usual performance measure for a tracking optimal control problem is:

$$J = \int_{\tau=0}^{\tau=t_f} [x(\tau) - r(\tau)]^T Q(\tau) [x(\tau) - r(\tau)] \, d\tau$$

where $Q(\tau)$ is a real, symmetric, positive semi-definite possible time-varying weighting matrix.

A regulator problem is a special case of a tracking problem in which the reference state r(t) is the origin of the state space, i.e., r(t) = 0. Any non-zero constant reference state can be converted to the state r(t) = 0 by a simple change of coordinates.

If the control inputs are bounded by constraints of the form:

$$|u_i(t)| \leq +1, \; i = 1, 2, \ldots, m \; ,$$

then the above performance measure is reasonable to use and the numerical method applied to solve the resulting two-point boundary value problem will produce a solution for the optimal control action $u^*(t)$ which automatically satisfies the control constraints.

If the control inputs for a tracking problem are not bounded, then minimizing the above performance measure will result in control inputs having impulse functions in their derivatives. To avoid this situation without placing arbitrary artificial bounds on the control inputs, a modified performance measure which includes a term depending on the control inputs is used:

$$J = \int_{\tau=0}^{\tau=t_f} \left\{ [x(\tau)-r(\tau)]^T Q(\tau) [x(\tau)-r(\tau)] + u(\tau)^T R(\tau) u(\tau) \right\} d\tau \; .$$

The weighting matrices $R(\tau)$ and $Q(\tau)$ are selected to trade off the relative importance of the state and control variables.

For optimal tracking control of a linear dynamic system, this modified performance measure leads to an easily-implemented optimal controller whose design is the solution to the linear quadratic control problem presented in the next subsection. This modified performance measure is also used when close tracking is desired and control energy is to be conserved.

The tracking problem as posed above makes no specific attempt to control the final state of the dynamic system. If the terminal state variable values are important, the above performance measure can be further modified by adding a term which explicitly depends on the terminal state:

$$J = [x(t_f) - r(t_f)]^T H [x(t_f) - r(t_f)] +$$

$$\int_{\tau=0}^{\tau=t_f} \left\{ [x(\tau) - r(\tau)]^T Q(\tau) [x(\tau) - r(\tau)] + u(\tau)^T R(\tau) u(\tau) \right\} d\tau \; .$$

Here, the matrix H is a real, symmetric, and positive semi-definite.

9.5.5 Linear Quadratic Optimal Control Problem

The linear quadratic optimal control problem is one particular optimal control problem for which it is possible to obtain a feedback controller. The optimal feedback controller will require the implementation of a set of time-varying gains. The optimal control input will be a linear function of these gains and the state variables, making its implementation either in analog hardware or a computer algorithm relatively simple.

The linear quadratic optimal control problem takes its name from the dynamic system to be controlled, a linear time-invariant continuous-time system, and the performance measure to be minimized, a quadratic function of the state and control variables. The dynamic system is described by:

$$\frac{dx(t)}{dt} = A(t)\,x(t) + B(t)\,u(t)\ .$$

In this mathematical model there are n state variables in the vector x and m control variables in the vector u. There is no requirement to specify the initial state of the dynamic system. The initial time is equal to 0 and the final time is T. The performance measure is:

$$J = x^T(T)\,S\,x(T) + \int_{\tau=0}^{\tau=T} \left[x^T(\tau)\,Q(\tau)\,x(\tau) + u^T(\tau)\,R(\tau)\,u(\tau)\right] d\tau\ .$$

The controller's objective is to drive the state x(t) as close as possible to the origin of the state space, while at the same time minimizing the control energy expended. The n by n matrix S assigns a weight to each component of the final state x(T). The n by n time-varying weighting matrix Q(t) and the m by m time-varying weighting matrix R(t) assign relative weights to the state trajectory followed and the control input applied over the interval of control. The weighting matrices are real, symmetric, positive semi-definite matrices whose elements are selected by the designer to provide a positive weight for each cross-product and squared term in the performance measure.

There are two cases of importance depending on whether the final time T is finite or infinite. For a finite final time the closed-loop optimal control $u^*(x(t))$ is given by:

$$u*(x(t)) = -R^{-1}(t)\,b^T(t)\,w(t)\,x(t)$$

where W(t) is the time-varying solution of the matrix Ricatti equation:

$$\frac{dW(t)}{dt} = -A^T(t)\,W(t) - W(t)\,A(t) - Q(t) + W^T(t)\,B(t)\,R^{-1}(t)\,B^T(t)\,W(t)$$

with the boundary condition W(T) = S. Note that this boundary condition is applied to W(t) at the terminal or final time of the control interval, and given the numerical values contained in the matrices A(t), B(t), R(t) and Q(t) the matrix W(t) can be found by integrating in reverse time. The matrix product $R^{-1}(t)B^T(t)W(t)$ can be interpreted as a time-varying feedback gain.

When the matrices A, B, Q, and R are constant and the matrix S equals a zero matrix, a solution for the case of an infinite time of control can be obtained:

$$u^*(t) = -R^{-1}\,B^T\,W\,x(t)\,.$$

Here, W is the solution of the algebraic matrix Ricatti equation:

$$0 = -A^T W - WA - Q + W^T BR^{-1}\,B^T\,W\,.$$

This result can be obtained by letting the matrix derivative dW(t)/dt equal zero, indicating a steady-state solution for the matrix W(t). Note that the optimal controller's feedback gains are no longer functions of time since the matrices R, B, and W are constant. In many linear quadratic control problems, the time-varying feedback gains which result in the finite time-of-control case will be seen to be very nearly constant until the time of control, T, has nearly expired. A commonly used approximation which avoids the need to compute and mechanize these time-varying gains is to solve for and use the gains determined for infinite time of control, accepting any performance degradation which may result.

As an example of the design of a closed-loop control system by means of the linear quadratic optimal control method, consider a first-order control system modeled by the state transition equation:

$$\frac{dx(t)}{dt} = -10\,x(t) + u(t)\,.$$

The performance measure to be minimized will be taken as:

$$J = \frac{1}{2}\,x^2(T) + \int_{\tau=0}^{\tau=T} \frac{1}{4}\,x^2(\tau) + \frac{1}{2}\,u^2(\tau)\,d\tau\,.$$

This performance measure assigns a weight of 1/4 to the state trajectory x(t), a weight of 1/2 to the control trajectory u(t) and a weight of 1/2 to the final state x(T). The final time T is specified to be 0.4 seconds. The state variable x(t) and the control action u(t) are not constrained in any way.

The time-varying closed-loop optimal control $u^*(x(t))$ is given by:

$$u^*(x(t)) = -R^{-1}(t)\, B^T(t)\, W(t)\, x(t)\ ,$$

where W(t) is the time-varying solution of the scalar Ricatti differential equation:

$$\frac{dW(t)}{dt} = -A^T W(t) - W(t)\, A - Q + W^T(t)\, BR^{-1} B^T W(t)\ .$$

with the boundary condition W(T) = S. In this example A = −10, B = +1, S = 1/2, Q = 1/4 and R = 1/2. The scalar Ricatti differential equation becomes:

$$\frac{dW(t)}{dt} = -2(-10)\, W(t) - \frac{1}{4} + W^2(t)\, (+1)^2\, \big(1/(1/2)\big),\ \text{or}$$

$$\frac{dW(t)}{dt} = 20\, W(t) - \frac{1}{4} + 2\, W^2(t),\ \text{with}\ W(0.4) = \frac{1}{2}\ .$$

Figure 9-5 shows the resulting state and control trajectories for an initial state of x(0) = 1.0, computed using rectangular integration with a time step of 0.004 seconds. The time-varying feedback gain $R^{-1}(t)B^T(t)W(t)$ is plotted in Figure 9-6. Note that the feedback gain increases as the time remaining for control decreases. This result is typical for any linear quadratic optimal control problem.

The solution for an infinite time of control can be obtained and applied to produce a suboptimal but non-time-varying controller:

$$u*(x) = -R^{-1}\, B^T\, W\, x\ .$$

Here, W is the solution of the algebraic matrix Ricatti equation:

$$0 = -A^T W - WA - Q + W^T\, BR^{-1}\, B^T\, W\ ,$$

which becomes

$$0 = -2(-10)W - (1/4) + W^2\, (1)^2\, \big(1/(1/2)\big)$$

or

$$0 = 2W^2 + 20W - (1/4)\ .$$

This is a simple quadratic whose positive root is W = 0.0125. Figure 9-7 is a plot of the state and control variable trajectories for the above dynamic system when the suboptimal steady-state feedback gain is used in place of the optimal time-varying gain. Note that the performance is not substantially different than that obtained using the optimal control policy.

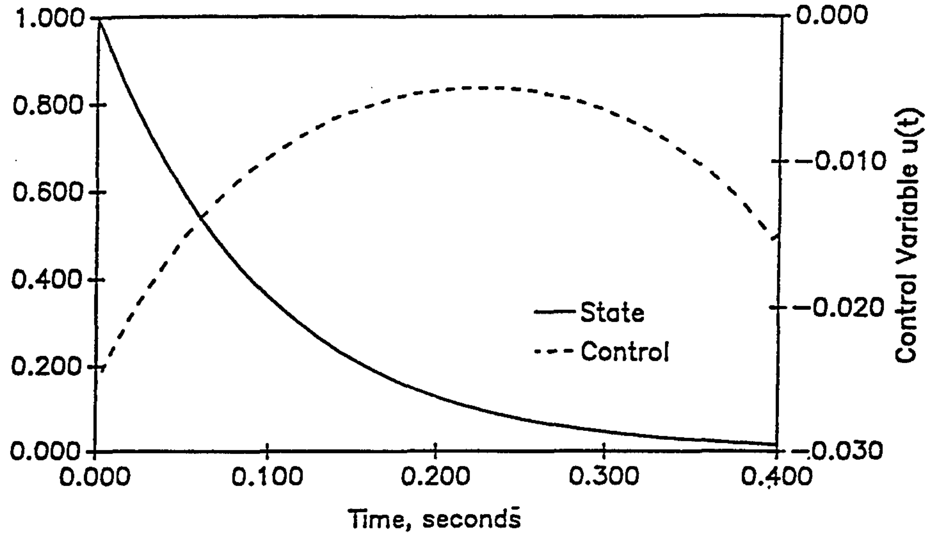


Figure 9-5. Optimal state and control variable trajectories.

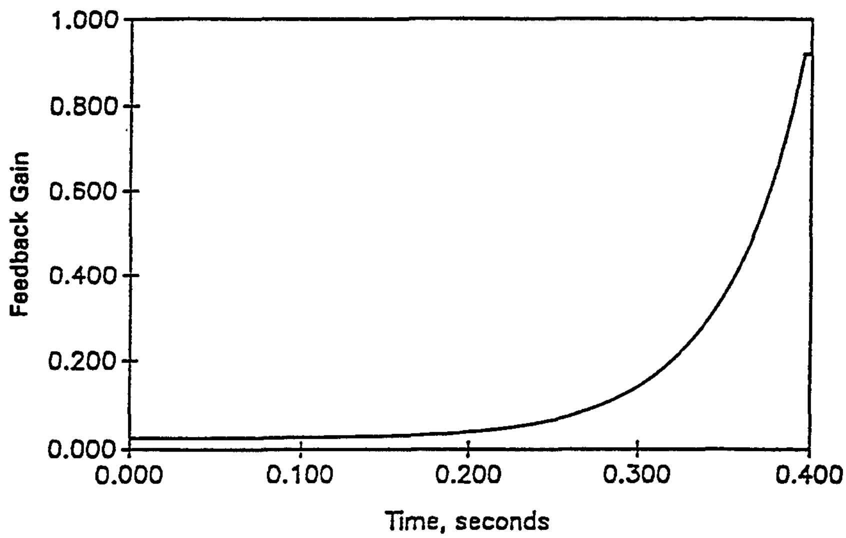


Figure 9-6. Time-varying feedback gain.

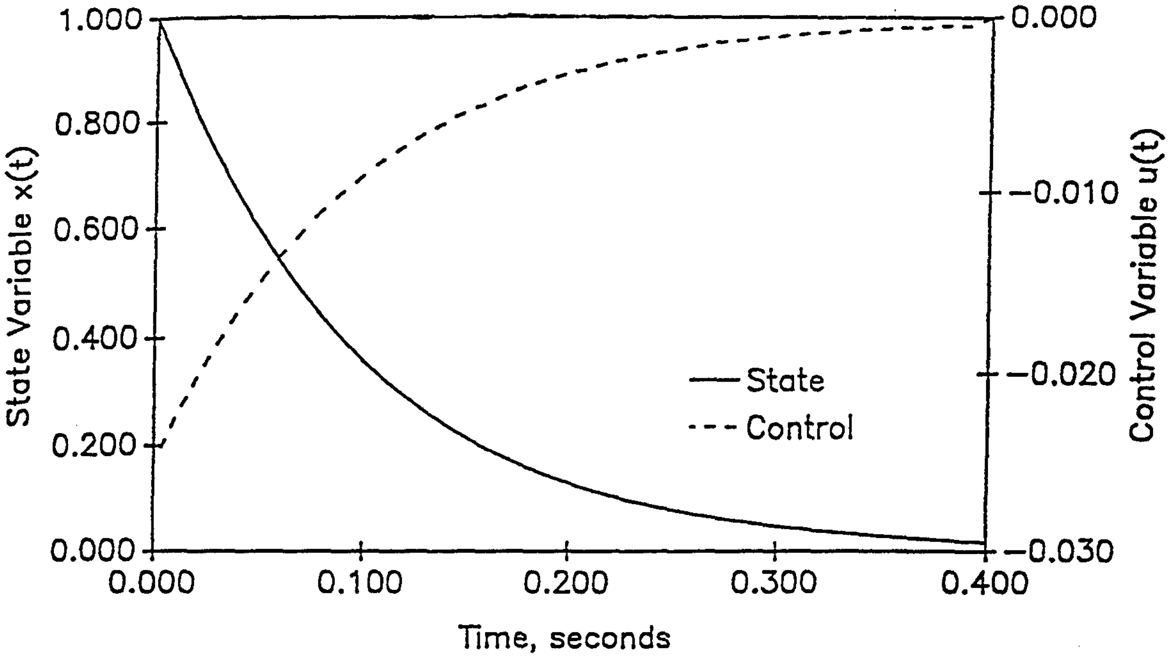


Figure 9-7. State and control variable trajectories using suboptimal steady-state feedback gain.

9.6 Summary

Optimal control theory involves the selection of a particular control input into a differential equation that models a dynamic system. Applications may include deterministic discrete-time dynamic systems or use of stochastic control with a random state transition process. One of the types of problems where optimal control theory was applied to a linear, time-varying case is called the time-optimal control problem. Pontryagin's maximum principle for solving this was described for the linear time-optimal control problem and for a second-order linear time-invariant system. A series of general optimal control problems were also discussed: minimum time; terminal control; minimum control effort; tracking and regulator; and linear quadratic optimal control. Feedback gains, which are usually difficult to express using optimal control was illustrated for the linear quadratic case.

REFERENCES

9.1 Bellman, R.E., Glicksburg, I., and Gross, O.A., On the "Bang-Bang" Control Problem, Quarterly of Applied Mathematics, Vol.14, pp.11-18, 1956.

9.2 Gamkrelidze, R.V., The Theory of Time Optimal Processes for Linear Systems, Izv. Adad. Nauk., SSR., Ser. Mat. 22, pp.449-474, 1958.

9.3 Krasovskii, N.N., On the Theory of Optimal Control, Automation and Remote Control, Vol.18, pp.1005-1016, 1957.

9.4 LaSalle, J.P., Boltayanski, V.G., and Gamkrelidze, R.V., Optimal Regulation Processes, Usp. Mat. Nauk. 14, pp.3-20, 1959.

9.5 Filippov, A.F., On Certain Questions on the Theory of Optimal Control, SIAM Journal of Control, Vol.1, pp.76-84, 1959.

9.6 Pontryagin, L.L. Boltyanski, V.G., and Gamkrelidze, R.V., Optimal Regulation Processes, Usp. Mat. Nauk. 14, pp.3-20, 1959.

9.7 Koo, D., Elements of Optimization, With Applications to Economics and Business, Springer-Verlag, New York, 1977.

9.8 Kirk, D.E., Optimal Control Theory, An Introduction, Prentice Hall, Inc., Englewood Cliffs, New Jersey, 1970.

This page is intentionally blank.

CHAPTER 10
SINGULAR PERTURBATION METHODS

10.1 Introduction to the Singular Perturbation Method

Singular perturbation methods are used to simplify the modeling, analysis, and design of controllers for high-order dynamic systems having both slow and fast dynamics. The method is physically motivated, based on the observation that some dynamic systems have states which can be divided into two classes, fast and slow. Membership of a state in either class is determined by the relative speed of its transient response. A change of time scale is used to facilitate the decomposition of the original system into these two subsystems, and this aspect is of interest since it preserves the structure of the underlying dynamic system.

Singular perturbation methods have been applied to a variety of problems including the stability analysis of linear and nonlinear continuous- and discrete-time systems, deterministic and stochastic optimal control, and linear and nonlinear filtering and estimation. In this chapter the basic method is introduced and several applications of the singular perturbation method are outlined.

The singular perturbation method has been detailed by Kokotovic et. al[10.1]. The singularly perturbed dynamic system is partitioned into two separate reduced-order dynamic subsystems. Each of these subsystems is assumed to evolve according to its own time scale. The basic singular perturbation approach relies on two fundamental assumptions:

(a) the dynamic subsystem having the slower transient response is obtained by assuming that the fast state variables all react instantaneously to a change in the slow state variables,

(b) the dynamic subsystem having the faster transient response is obtained by assuming that the slow state variables all remain constant during any transient responses of the fast state variables.

The dynamic system of interest is modeled by two sets of state transition differential equations:

$$\frac{dx(t)}{dt} = f\big(x(t),\ y(t),\ t,\ \epsilon\big)\ ,$$

$$\epsilon\,\frac{dy(t)}{dt} = g\big(x(t),\ y(t),\ t,\ \epsilon\big)\ .$$

In these equations $x(t)$ is an n-dimensional state vector and $y(t)$ is an m-dimensional state vector. The states which comprise $x(t)$ are called the slow state variables and those which comprise $y(t)$ are called the fast variables. The parameter ϵ is assumed to be a small positive number.

This system is called a singularly perturbed system because the parameter ϵ multiplies the vector of derivatives $dy(t)/dt$, and setting ϵ to a value of zero reduces the order of the dynamic system from $(n+m)$ to n. For small numerical values of the parameter ϵ the time derivative $dy(t)/dt = g(x(t), y(t), t, \epsilon)/\epsilon$ is large as long as the function $g(.)$ is not equal to zero, and so $y(t)$ changes rapidly compared to $x(t)$.

One way to investigate the operation of this dynamic system and understand the singular perturbation methodology is to consider an initial value problem in which the initial conditions $x(0) = x_0$ and $y(0) = y_0$ are specified. To study the behavior of the singularly perturbed dynamic system, one can set the parameter ϵ to a numerical value of zero, and strive for an approximate solution to the resulting system of differential and algebraic equations. The dynamic system having ϵ set to zero is called the reduced system and is mathematically given by:

$$\frac{dx'(t)}{dt} = f\big(x'(t), y'(t), t, 0\big), \; x'(0) = x_0 \, ,$$

$$0 = g\big(x'(t), y'(t), t, 0\big) \, .$$

This reduced system is of order n, since it is described by the set of n differential equations for the slow variables $x'(t)$. The states $x'(t)$ and $y'(t)$ are considered to be approximations of the true states $x(t)$ and $y(t)$. Only n of the specified initial conditions (x_0, y_0) can be satisfied by the reduced system of equations. The easiest way to accomplish this is to enforce the n initial conditions on the slow variables $x(t)$, and solve the resulting n first-order differential equations.

The method of singular perturbations is limited to those dynamic systems where the roots, or solutions, of the algebraic equation of the reduced system are real and distinct, and the partial derivative of $g(.)$ with respect to y' is nonsingular. Let a root of this algebraic equation be given by:

$$y'(t) = h\big(x'(t), t\big) \, .$$

This gives an expression for $y'(t)$ in terms of $x'(t)$. Substitute this expression back into the differential equation for the slow variables $x'(t)$:

$$\frac{dx'(t)}{dt} = f\Big(x'(t), h\big(x'(t), t\big), t, 0\Big), x'(0) = x_0' \, .$$

In performing this operation it is hoped that $x'(t)$ will be a good approximation to the true solution $x(t)$ for all values of time t, and that $y'(t)$ will be a good approximation to the true solution $y(t)$ for all values of time t except near $t = 0$, the starting time, since the initial condition on the fast variables $y(t)$ has not been enforced and the approximation $y'(0)$ is usually not equal to the required initial value of the fast variable $y(0) = y_0$.

To further investigate the behavior and solution for $y(t)$ near $t = 0$, the time scale is stretched by applying a scaling transformation:

$$\tau = \frac{t}{\epsilon}, \text{ so that}$$

$$d\tau = \frac{dt}{\epsilon} \text{ or}$$

$$dt = \epsilon\, d\tau .$$

Using this transformation in the original dynamic system equations, one obtains:

$$\frac{dx(\tau)}{d\tau} = \epsilon f\left(x(\tau),\ y(\tau),\ \epsilon\tau,\ \epsilon\right),\ x(0) = x_0 ,$$

$$\frac{dy(\tau)}{d\tau} = g\left(x(\tau),\ y(\tau),\ e\tau,\ e\right),\ y(0) = y_0 .$$

Setting the parameter ϵ equal to zero in these equations produces the initial condition on the slow variables since the derivative of x with respect to τ is now zero:

$$\frac{dx(\tau)}{d\tau} = 0 ,$$

$$x(\tau) = x_0 .$$

The error between the true and approximate solution for the fast variables $y(t)$ can now be defined and a differential equation for this error η obtained by taking a derivative with respect to the stretched time τ:

$$\eta(\tau) = y(\tau) - y'(\tau) , \text{ so that}$$

$$\frac{d\eta(\tau)}{d\tau} = \frac{dy(\tau)}{d\tau} - \frac{dy'(\tau)}{d\tau} ,$$

$$= g\left(x(\tau),\ y(\tau),\ \epsilon\tau,\ \epsilon\right) - \frac{dh\left(x'(t),\ t\right)}{d\tau} ,$$

$$= g\left(x_0,\ y'(0) + \eta(\tau),\ 0,\ 0\right) ,$$

with the initial condition:

$$\eta(0) = y_0 - y(0) .$$

The vector $\epsilon(\tau)$ represents the pure fast part of the vector y(t). This system is called the boundary layer system.

The fundamental result in the singular perturbation methodology is that, under certain technical conditions[10-1], the solution to the initial value problem for the original dynamic system modeled by the slow variables x(t) and the fast variables y(t) can be replaced, in the limit as the parameter ϵ approaches zero, by the simultaneous solution of the reduced and boundary layer problems, and that the approximate solution will agree numerically with the exact solution to within a factor having a magnitude on the order of the parameter ϵ:

$$x(t) = x'(t) + 0(\epsilon) ,$$

$$y(t) = y'(t) + \epsilon(\tau) + 0(\epsilon) .$$

To apply the singular perturbation method to the design of a control system, the designer begins by defining the reduced and boundary layer problems in terms of the mathematical model of the original system. Insight and intuition must be used to define the vectors of fast and slow variables. The approximate solution to the original problem is assumed to result from the combination of the solutions to the reduced and boundary layer problems.

A simple example will be used to illustrate the methodology. A dynamic system having both slow and fast dynamics is described by the following state transition equations:

$$\frac{dx(t)}{dt} = -x(t) + y(t) + u(t),\ x(0) = x_0 ,$$

$$\epsilon \frac{dy(t)}{dt} = -x(t) + y(t) + u(t),\ y(0) = y_0 .$$

The output is given by:

$$z(t) = x(t) + 2.0\ y(t) .$$

The initial conditions are $x_0 = 0.0$ and $y_0 = 0.0$ and the input u(t) is a unit step function, $u(t) = 1.0$. The underlying dynamic system is of order two, and a reduced system of order one is obtained by setting the parameter ϵ equal to zero:

$$\frac{dx'(t)}{dt} = -x'(t) + y'(t) + u(t),\ x'(0) = x_0 ,$$

$$0 = -x'(t) + y'(t) + u(t) \ .$$

The algebraic equation can be solved for $y'(t)$:

$$y'(t) = -x'(t) + u(t) \ ,$$

and this result substituted into the differential equation:

$$\frac{dx'(t)}{dt} = -x'(t) + \left(-x'(t) + u(t)\right) + u(t) \ , \text{ or}$$

$$\frac{dx'(t)}{dt} = -2.0\ x'(t) + 2.0\ u(t),\ x'(0) = x_0 \ .$$

Note that $y'(0) = -x'(0) + u(0) = -x_0 + 1.0$ does not, in general, equal the specified initial condition y_0. The approximate solution $x'(t)$ is intended to be a good approximation to the slow variable $x(t)$ for all values of time, and the approximation $y'(t)$ is intended to be a good approximation to the fast variable $y(t)$ for all values of time except those near $t = 0.0$.

To investigate the behavior of the fast variable $y(t)$ near $t = 0.0$ the time scale is stretched by defining a new time τ:

$$\tau = \frac{t}{\epsilon} \ , \text{ so that}$$

$$d\tau = \frac{dt}{\epsilon} \ , \text{ and}$$

$$\frac{dx(\tau)}{d\tau} = \epsilon\left(-x(\tau)+y(\tau)+u(\tau)\right),\ x(0) = x_0 \ ,$$

and

$$\frac{dy(\tau)}{d\tau} = \left(-x(\tau)+y(\tau)+u(\tau)\right),\ y(0) = y_0 \ .$$

Setting the parameter ϵ equal to zero eliminates the first differential equation, and then $x(\tau) = x_0$.

The error between the true solution $y(t)$ and the approximate solution $y'(t)$ can be written as:

$$\eta(\tau) = y(\tau) - y'(\tau)$$

and a differential equation for the error developed:

$$\frac{d\eta(\tau)}{d\tau} = -x_0 - \left(y'(0)+\eta(\tau)\right) + u(0) \ , \text{ or}$$

$$\frac{d\eta(\tau)}{d\tau} = -x_0 - \left(-x'(0)+u(0)+\eta(\tau)\right) + u(0) \text{ , or}$$

$$\frac{d\eta(\tau)}{d\tau} = -\eta(\tau) \text{ , with the initial condition:}$$

$$\eta(0) = y_0 - \left(-x_0+1.0\right) = y_0 + x_0 + 1.0 \text{ .}$$

The approximate solution to this problem is then obtained by collecting all of the above results:

$$x(t) \approx x'(t) + 0(\epsilon) \text{ , with}$$

$$\frac{dx'(t)}{dt} = -2.0\,x'(t) + 2.0\,u(t) \text{ ,}$$

$$x'(0) = x_0,\ u(t) = 1.0 \text{ ,}$$

$$\frac{d\eta(t)}{dt} = -\frac{\eta(t)}{\epsilon} \text{ , with}$$

$$\epsilon(0) = y_0 + x_0 - u(0) \text{ ,}$$

$$y'(t) = -x'(t) + u(t) \text{ ,}$$

and finally:

$$y(t) \approx y'(t) + \eta(t) + 0(\epsilon) \text{ .}$$

The parameter ϵ may be set to 0.001 in this example. In this case, the exact and approximate solutions for the variable x(t) are essentially identical for all values of time t. Note that the approximate solution obtained for the fast variable y'(t) is very good, especially near t = 0.0 and again for large values of time, when compared with the exact solution for the variable y(t). The results of the exact and approximate outputs z(t) and z'(t) are very good, and the influence of the approximate nature of y'(t) can be seen in the approximate output z'(t).

10.2 Stability Analysis of Singularly Perturbed Systems

Saberi and Khalil[10.2] have investigated the stability of a nonlinear singularly perturbed system described by the following set of state transition differential equations:

$$\frac{dx(t)}{dt} = f\left(x(t),\ y(t)\right) \text{ ,}$$

$$\epsilon\,\frac{dy(t)}{dt} = g\left(x(t),\ y(t)\right) \text{ .}$$

Their results are based on the application of Lyapunov's method and take the form of a set of mathematical conditions on the functions f(.) and g(.) which, if satisfied, guarantee asymptotic stability of the underlying dynamic system.

For a linear time-invariant dynamic system represented by:

$$\frac{dx(t)}{dt} = A_{11}\, x(t) + A_{12}\, y(t) ,$$

$$\epsilon \frac{dy(t)}{dt} = A_{21}\, x(t) + A_{22}\, y(t) ,$$

their results reduce to a set of requirements on the real parts of the eigenvalues of the underlying dynamic system:

$$\Re \left\{\lambda\left(A_{11} - A_{12} A_{22}^{-1} A_{21}\right)\right\} > 0.0 ,$$

$$\Re \left\{\lambda\left(A_{22}\right)\right\} > 0.0 .$$

10.3 Optimal Control for Singularly Perturbed Systems

The singular perturbation method has been applied to the optimal control of a nonlinear dynamic system described by the following set of state transition differential equations:

$$\frac{dx(t)}{dt} = f\left(x(t),\ y(t),\ u(t),\ t,\ \epsilon\right),\ x(0) = x_0 ,$$

$$\epsilon \frac{dy(t)}{dt} = g\left(x(t),\ y(t),\ u(t),\ t,\ \epsilon\right),\ y(0) = y_0 ,$$

and the performance measure:

$$\text{minimize}\ \ J = \int_{\tau=0}^{\tau=T} V\left(x(\tau),\ y(\tau),\ u(\tau),\ \tau,\ \epsilon\right) d\tau ,$$

by Freedman and Kaplan[10.3] and O'Malley[10.4].

Necessary conditions which must be satisfied by the optimal control action u(t) were developed by applying Pontryagin's maximum principle. A Hamiltonian was first defined:

$$H = \begin{bmatrix} V\left(x(t),\ y(t),\ u(t),\ t,\ \epsilon\right) \\ +\ p^T(t)\ f\left(x(t),\ y(t),\ u(t),\ t,\ \epsilon\right) \\ +\ \epsilon\ q^T(t)\ g\left(x(t),\ y(t),\ u(t)\ t,\ \epsilon\right) \end{bmatrix} ,$$

where p(t) and q(t) are the costates for the slow variables x(t) and the fast variables y(t). Application of optimal control theory and Pontryagin's maximum principle then yields a set of necessary conditions which must be satisfied by the optimal control action:

$$\frac{dx(t)}{dt} = f\left(x(t),\ y(t),\ u(t),\ t,\ \epsilon\right),$$

$$\epsilon\frac{dy(t)}{dt} = g\left(x(t),\ y(t),\ u(t),\ t,\ \epsilon\right),$$

$$\frac{dp(t)}{dt} = -\frac{\partial H}{\partial x},$$

$$\epsilon\frac{dq(t)}{dt} = -\frac{\partial H}{\partial y}, \text{ and}$$

$$\frac{\partial H}{\partial u} = 0 .$$

The boundary conditions for the resulting two-point boundary-value problem are:

$$x(0) = x_0 ,$$

$$y(0) = y_0 ,$$

$p(T) = 0$, since the final state x(T) is unspecified, and

$q(T) = 0$, since the final state y(T) is unspecified.

An approximate solution to this problem can be generated by means similar to those used for the stability analysis outlined above. A reduced system is first obtained, followed by the development of two coupled boundary-layer systems which account for the initial conditions on the variables y(t) and the terminal conditions on the costates. A solution for the optimal control action having the form:

$$u(t,\epsilon) = u_1(t) + u_2(\tau) + u_3(\sigma) + 0(\epsilon), \text{ where}$$

$$\tau = \frac{t}{\epsilon}, \text{ and}$$

$$\sigma = \frac{(T-t)}{\epsilon}$$

are obtained where $u_1(t)$ is the optimal solution for the reduced system, $u_2(\tau)$ is the optimal solution for the initial boundary-layer system and $u_3(\sigma)$ is the optimal solution for the terminal boundary-layer solution.

10.4 Application to Linear Quadratic Optimal Control

The linear quadratic optimal control problem defined by the linear time-invariant state transition equations:

$$\begin{bmatrix} \dfrac{dx(t)}{dt} \\ \dfrac{dy(t)}{dt} \end{bmatrix} = A \begin{bmatrix} x(t) \\ y(t) \end{bmatrix} + B[u(t)]$$

and in which the minimization of a quadratic performance measure:

$$J = \int_{\tau=0}^{\tau=T} \left\{ \begin{bmatrix} x(\tau) \\ y(\tau) \end{bmatrix}^T Q \begin{bmatrix} x(\tau) \\ y(\tau) \end{bmatrix} + [u(\tau)]^T R[u(\tau)] \right\} d\tau$$

is required has been extensively and actively investigated by O'Malley[10.4], Yackel and Kokotovic[10.5] and others.

In this formulation the matrices A, B and Q are given by:

$$A = \begin{bmatrix} A_1 & A_2 \\ \dfrac{A_3}{\epsilon} & \dfrac{A_4}{\epsilon} \end{bmatrix},$$

$$B = \begin{bmatrix} B_1 \\ \dfrac{B_2}{\epsilon} \end{bmatrix}, \text{ and}$$

$$Q = \begin{bmatrix} Q_1 & Q_2 \\ Q_2^T & Q_3 \end{bmatrix}.$$

The matrix R must be positive definite and the matrix Q must be positive semi-definite. These matrices may also be time-varying, but A_4 must be non-singular (i.e., A_4 must possess an inverse) over the time of control.

The optimal solution to this linear quadratic optimal control problem is:

$$u(t) = -R^{-1}(t)\, B^T(t)\, K(t, \epsilon) \begin{bmatrix} x(t) \\ y(t) \end{bmatrix},$$

where the matrix $K(t, \epsilon)$ satisfies the Riccati differential equation:

$$\frac{dK(t,\epsilon)}{dt} = -Q(t) - K(t,\epsilon)\,A(t) - A^T(t)\,K(t,\epsilon)$$

$$+\,K(t,\epsilon)\,B(t)\,R^{-1}(t)\,B^T(t)\,K(T,\epsilon)\ ,$$

with a terminal boundary condition of $K(T, \epsilon) = 0$.

The complete solution to this problem requires an asymptotic analysis of the behavior of the elements of the matrix K(.) as the parameter ϵ approaches zero. As in the previous example, two sets of boundary-layer equations are introduced, one to account for the initial conditions y(0) and the other to account for the terminal conditions $K(T, \epsilon)$.

Much work has been devoted to the design of reduced-order feedback controllers similar to that outlined above. The implementation of a controller in state-feedback form may also require the design of a state variable estimator or observer, and this requirement can be computationally expensive for dynamic systems having many state variables. For that reason the development of output-feedback controllers for singularly-perturbed systems has also received substantial attention.

10.5 Summary

The development of any control system requires a mathematical model of the underlying dynamic system. Realistic representations of most systems utilize higher order differential equations in which numerous small-valued parameters, some often parasitic in nature and others associated with relatively small time constants, are involved. In the construction of a detailed model of the system's behavior, these parasitic effects cause the higher system order. If, when the effect of these small parameters is suppressed by setting their numerical values to zero, the resulting dynamic system is of lower order and the system is said to be singularly perturbed.

Singularly-perturbed dynamic systems arise naturally when investigating the dynamic response of complex electronic circuits, where the effects of various circuit elements are temporarily ignored by setting their numerical values to zero or infinity in a process which represents certain open or short-circuit conditions.

The use of the singular-perturbation method has, in many cases, resulted in an increased understanding of the underlying system dynamics and the development of efficient computational algorithms for the solution of stability, control, and optimization problems. Though mathematically complex, the theory of singular perturbations was one of the most active research areas of modern control theory[10.6] during the 1980's.

REFERENCES

10.1 Kokotovic, P. V., O'Malley, R. E., and Sannuti, P., Singular Perturbations and Order Reduction in Control Theory — An Overview, Automatica, Vol. 12, pp. 123-132, 1976.

10.2 Saberi, A and Khalil, H. K., Quadratic-Type Lyapunov Functions for Singularly Perturbed Systems, Proceedings of the 20th IEEE Conference on Decision and Control, Institute of Electrical and Electronic Engineers, New York, pp. 204-205, 1981.

10.3 Freedman, M. I. and Kaplan, J. L., Singular Perturbations of Two-Point Boundary-Value Problems Arising in Optimal Control, SIAM Journal on Control, Vol. 14, pp. 189-215, 1971.

10.4 O'Malley, Jr., R. E., Singular Perturbations and Optimal Control, In: Coppel, W. A., ed., Mathematical Control Theory, Springer-Verlag, New York, pp. 170-218, 1977.

10.5 Yackel, R. A. and Kokotovic, P. V., A Boundary Layer Method for the Matrix Ricatti Equation, IEEE Transactions on Automatic Control, Vol. TAC-18, pp. 17-24, 1973.

10.6 Saksena, V. R., O'Reilly, J., and Kokotovic, P. V., Singular Perturbations and Time-Scale Methods in Control Theory: Survey 1976-1983, Automatica, Vol. 20, pp. 273-293, 1984.

This page is intentionally blank.

CHAPTER 11
STOCHASTIC CONTROL

11.1 Formulating Stochastic Control Problems

This Chapter highlights several concepts and design approaches of stochastic optimal control theory. A stochastic optimal control problem involves the presence of significant uncertainties mainly due to random occurrences, not necessarily noise and device tolerances. Uncertainties which affect the successful operation of a tactical guided missile include unforseen target maneuvers, component and software defects, or failures and applied countermeasures.

In a deterministic optimal control problem, optimal open-loop control actions can be computed in advance, and, in certain special cases, optimal closed-loop control actions, which depend only on the state of the dynamic system, can be constructed. In a stochastic optimal control problem open-loop control actions which do not account for the presence of chance occurrences will result in suboptimal performance compared to a closed-loop feedback control action which incorporates information available as the dynamic system state evolves over time.

For deterministic dynamic systems there are two basic theoretical approaches toward the development of controllers. Both approaches begin with the development of a mathematical model for the underlying dynamic system. The first approach, based on stability theory, assumes that the form of the controller has been specified by the designer. The controller parameters are then selected to ensure the stability of the controlled system. Examples of this approach were presented in Chapter 7 of this report.

The second approach toward the development of a controller for a deterministic dynamic system, the use of optimal control theory, requires the system designer to specify a performance measure for the controlled dynamic system. The designer then attempts to compute a control policy which optimizes the performance measure and simultaneously satisfies the state transition equation and any other constraints applied to the mathematical model. Examples of the optimal control approach were presented in Chapter 9 of this report.

Real-world control problems never exactly conform to the mathematical model used to design a controller by either deterministic approach. As a result, the stability margin or optimal performance computed when the system is designed is hardly ever realized in practice. Consequently, the major application of the optimal control theory approach is often not to design an optimal system, but rather to use optimal control theory as a mathematical tool for organizing the control system design process. Optimal control theory can yield design insights regarding control system structures and performance limits, insights that may be obscure if the control system is designed in an ad hoc, heuristic manner.

When uncertainty enters into the control problem, the designer enters the realm of stochastic optimal control theory. Stochastic optimal control is concerned with mathematical questions regarding the best manner in which to control an uncertain dynamic system. The uncertainty may arise from measurement errors, severe noise effects, lack of a precise mathematical model or other sources such as manufacturing tolerances.

The term stochastic process refers to a dynamic system whose evolution over time is influenced by a set of random variables or disturbances. Although some knowledge regarding the statistical nature of these disturbances is assumed to be available to the designer, the behavior of the stochastic process cannot be predicted exactly, since the system state and output are random variables. Statistical measures must then be used to describe the system's evolution. For example, one may be able to predict the form of the probability density function for and the expected value of a state variable in a stochastic process, rather than the precise value of the state after the passage of time.

A stochastic optimal control problem is specified by a state-transition mechanism, a set of admissible control inputs, and a performance measure, J, which assigns a numerical value to the use of a particular control policy. The state transition mechanism may be a continuous-time differential equation, a discrete-time difference equation or a state transition table which indicates the probability of the next state and output for each present state and applied control action. The state variables and applied control actions may be limited in magnitude or otherwise constrained, the initial state may be specified precisely or, more likely, described by a probability distribution function, the desired final state may or may not be specified, and the time allowed for control may be finite, infinite, or described implicitly by the first occurrence of some system state or output.

Since the dynamic system is affected by not only the applied control action but also by random disturbances, the use of one specific control action will not generally produce a repeatable result in terms of a state trajectory. Rather, a set of trajectories will be obtained during a set of repeated experiments, and from this set of trajectories a probability density function which describes the observed performance given the specific control action can be evaluated. For this reason the performance measure in a stochastic control problem is usually stated in terms of an expected value $E\{J\}$.

A continuous-time stochastic process, for example, might be represented by the following set of equations:

$$\frac{dx(t)}{dt} = A(t)\,x(t) + B(t)\,u(t) + w(t)\,,$$

$$E\{x(0)\} = x_0\,.$$

The differential equation models a conventional linear time-varying dynamic system in which x(t) is the state of the system at time t, u(t) is an applied control action, and w(t) is a white Gaussian noise process having a specified mean value vector and covariance matrix. A diagram of this process is indicated in Figure 11-1. This noise process may represent the effect of an external disturbance, such as jamming, on the behavior of the dynamic system. The initial state of the dynamic system is specified here in terms of its expected value, rather than in terms of a precise initial value, and the probability density function for the initial state is assumed to be available.

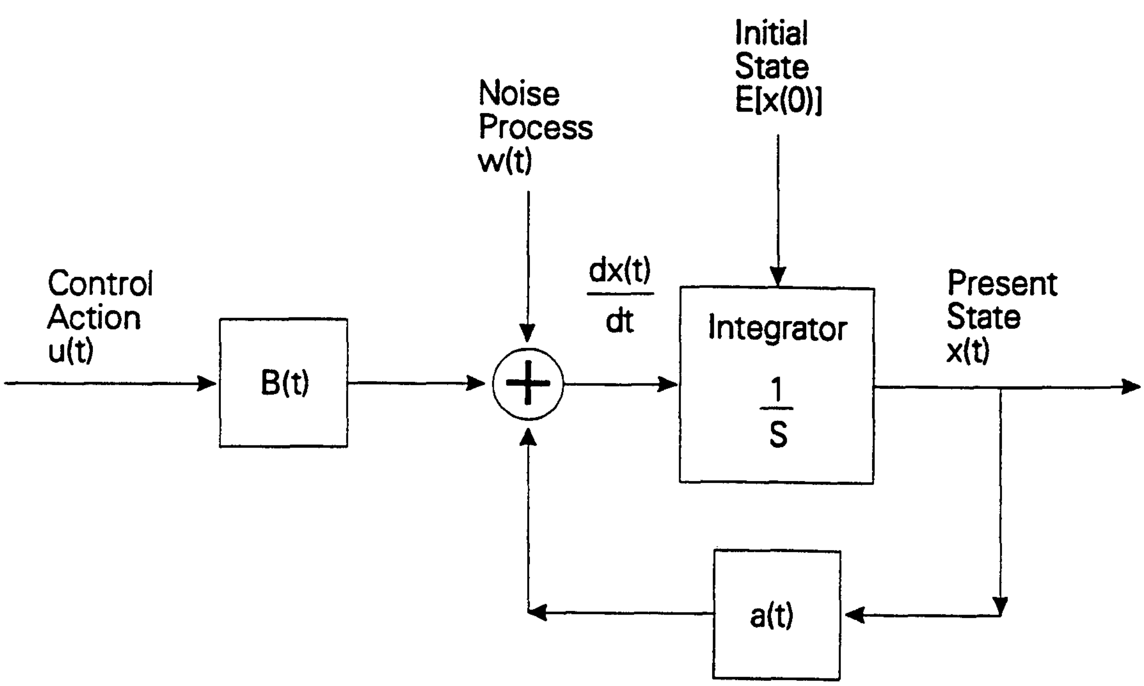


Figure 11-1. A continuous-time stochastic process.

If the operation of this system were simulated by means of repeated trials and the application of the same control action for each trial, a family of state trajectories would be generated, depending on the initial state randomly selected at the start of each trial and the noise process generated over the duration of each trial.

A discrete-time stochastic process (Figure 11-2) might be represented in a similar manner by the following set of equations:

$$x(k+1) = A(k)x(k) + B(k)u(k) + w(k) ,$$

$$E\{x(0)\} = x_0 .$$

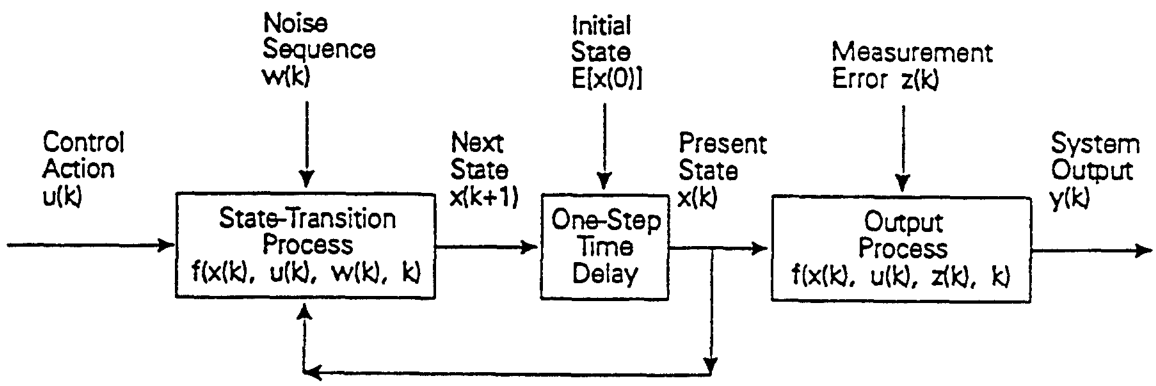


Figure 11-2. A discrete-time stochastic process.

The difference equation models a discrete-time, linear, time-varying dynamic system in which x(k) is the state of the system at time k, u(k) is an applied control action and w(k) is a white Gaussian noise sequence having a specified mean value vector and covariance matrix. This noise sequence may represent the effect of an external disturbance on the dynamic system's behavior. The initial system state is again specified in terms of its expected value and probability density function.

If the operation of this discrete-time system were simulated by means of repeated trials and the application of the same control action for each trial, a family of discrete-time state trajectories would be generated, each depending on the initial state randomly selected at the start of each trial and the generated noise sequence. Figure 11-3 shows the resulting series of random process trajectories.

The objective in controlling a discrete-time system such as the one described above is the optimization of some expected measure of performance, for example:

$$\text{minimize } E\{J\} = E\left\{ x^T(K)Sx(K) + \sum_{k=0}^{k=K-1} \left[x^T(k)Qx(k) + u^T(k)Ru(k)\right] \right\} .$$

The value of this performance measure computed over each simulated trial would be different due to the presence of the noise process and the random initial state, and the accumulated results could be used to develop a probability density function for the performance measure J which would

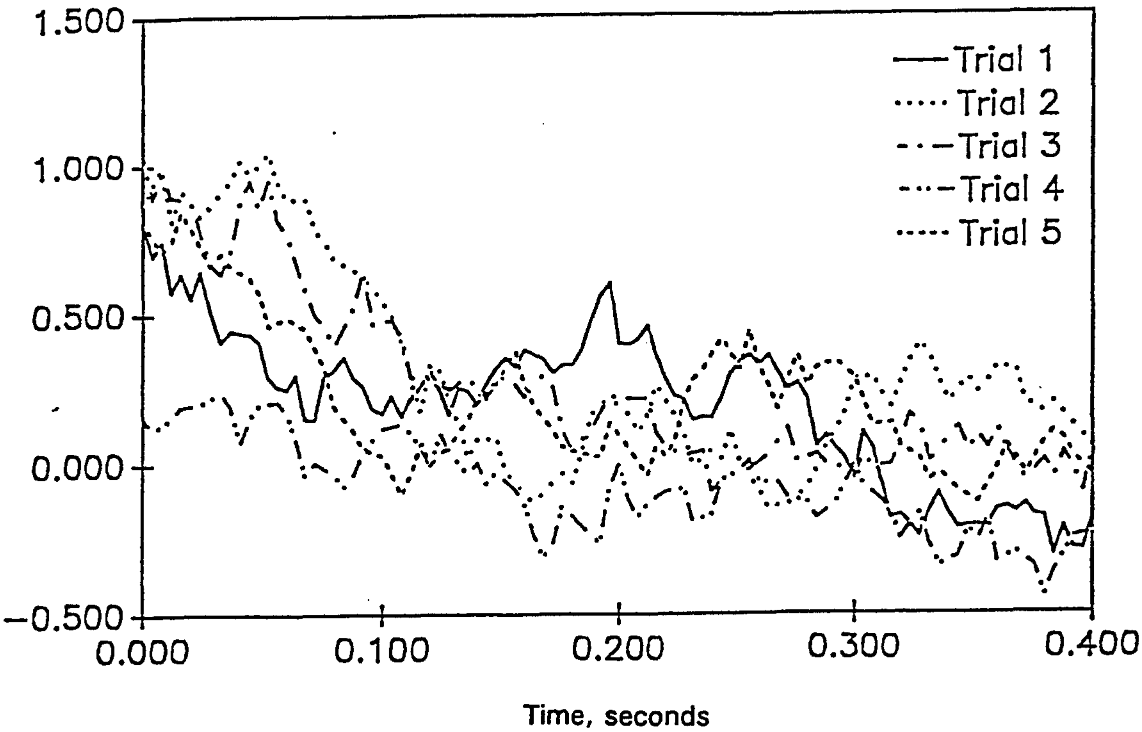


Figure 11-3. Random process state trajectories.

depend on the control action selected and applied over all trials. The expected value of J given the control action selected, E{J}, could then be computed.

By repeating this experiment for a variety of applied control actions, an open-loop control action which yields an apparent minimum value of E{J} can be determined. The difficulty with this Monte Carlo approach is that a guaranteed global minimum of the performance measure E{J} is not assured, a large number of repeated trials must be performed, and there is no way to determine in advance what the mathematical form of the optimal control action $u^*(k)$ should be.

11.2 State Transition Tables and Diagrams

Stochastic processes described by tabulated state transition mechanisms serve as useful models for a variety of simple control problems, and these models may be extended to the control of more complicated discrete-time dynamic systems in which the states and controls are constrained to finite (integer) sets. The resulting models are similar to the state transition models encountered in the design of finite-state logic machines.

For example, a dynamic system may be defined by a set of states which can take on the integer values 0, 1, 2, or 3. These states might represent the observed output of a simple counter. The applied control input might be restricted to the values 0, +1, or −1, representing an actuating signal to hold the present count, count up one value, or count down one value. The operation of the counter might be corrupted in some way such that the counter occasionally functions in the wrong manner. By observing the operation of the counter over a long period of time and trying all of the available control inputs, a set of state transition probabilities can be developed which depend on the present state and the applied control input. For example, when the counter is in state 1, and the count up command u = +1 is applied, 70% of the time the next state is +2, indicating a correct operation, and 30% of the time the next state is 0, indication an incorrect operation.

The state transition process for this system can be described as in Table 11-1 which indicates the present state, the applied control input, and the probability of attaining each of the next states:

TABLE 11-1. STATE TRANSITION PROBABILITIES FOR A STOCHASTIC OPTIMAL CONTROL SYSTEM

Present State	−0				0				+1			
	0	1	2	3	0	1	2	3	0	1	2	3
0	-	-	-	1.0	1.0	-	-	-	-	1.0	-	-
1	0.9	-	0.1	-	-	1.0	-	-	0.3	-	0.7	-
2	-	1.0	-	-	-	-	1.0	-	-	-	-	1.0
3	0.2	-	0.8	-	-	-	-	1.0	0.6	-	0.4	-

A performance measure can be defined for a discrete-time stochastic system. The cost of operating the system over each transition is usually a function of the state of the system at the start of each transition and the control action applied. The performance measure is computed as the expected value of a sum of costs incurred over the sequence of state transitions and possibly the terminal system state. In a problem of this type the selection of a control strategy is equivalent to the selection of a set of state transition probabilities.

The operation of a discrete-time, discrete-state stochastic system can be investigated by first designing a control strategy which specifies the state transition probabilities in terms of each possible present state and control input. Once this is done the operation can be simulated in detail by beginning at an initial state, applying a control input, randomly determining the next state, accumulating the incurred cost over the transition, and repeating this process for the number of transitions necessary as specified by the control interval. The resulting set of state and control histories and the accumulated performance measure values can be used to estimate the probability distribution function for the performance measure J, and its expected value.

The goal in applying optimal stochastic control theory to dynamic systems such as those illustrated above is to automatically determine, by means of a mathematical procedure, the nature of the optimal control action which yields an optimum expected value of the performance measure and satisfies any constraints imposed on the solution of the problem. The solution of these problems is, in general, an area of active research. While solutions are known for a limited class of problems, much work remains to be done[11.1]. Techniques which have been developed in stochastic control theory have also been applied in other application areas such as adaptive control.

11.3 Feedback in Stochastic Control

The notions of feedback are somewhat different in the case of a deterministic optimal control problem and that of a stochastic optimal control problem. In a deterministic optimal control problem the optimal control action can be computed in advance and implemented as either an open- or closed-loop control policy. There will be no difference in terms of the performance measure if either form of the control action is used since the response of the deterministic system is completely specified once the system dynamics, the initial state, and the control action is specified. A feedback controller implemented in a deterministic control application does not yield a lower value of the performance measure when compared to an open-loop controller, since the system input and output are completely determined. The deterministic optimal control action can be computed and applied as either an open-loop or a closed-loop function.

In a stochastic optimal control problem it is always better to implement a closed-loop control policy which measures the state or output of the system and then determines an appropriate control action. This process takes advantage of all available information about the operation of the stochastic process as it becomes available.

A simple example can be used to illustrate the advantage of a feedback controller for a stochastic system. Consider a single-stage discrete-time, stochastic control problem with a state transition mechanism specified by a simple difference equation:

$$x_1 = x_0 + u_0 .$$

The initial state value x_0 takes on the integer values 0 or 1 with an equal probability of being in either state. This means that on each repeated trial the system will initially be observed to be in either state 0 or state 1, and in the long run will be found in both states an equal number of times. The expected value of the state x_0 is thus 0.50.

In this example the performance measure to be minimized is:

$$J = E\left\{x_1^2\right\} .$$

The optimal open-loop control action is to select u_0 equal to $-1/2$ at the start of each trial. This can be seen by determining the expected value of x_1^2:

$$E\left\{x_1^2\right\} = E\left\{(x_0 + u_0)^2\right\}$$

$$= E\left\{x_0^2\right\} + 2E\left\{x_0 u_0\right\} + E\left\{u_0^2\right\}$$

$$= \left[0.5(0)^2 + 0.5(1)^2\right] + 2u_0\left[0.5(0) + 0.5(1)\right] + E\left\{u_0^2\right\}$$

$$= 0.5 + 2u_0[0.5] + u_0^2$$

$$E\left\{x_1^2\right\} = 0.5 + u_0 + u_0^2 .$$

To minimize this expected value, take a derivative of the expected value with respect to u_0 and set the result to zero and solve for the required value of u_0:

$$0 = 1 + 2u_0 , \text{ or } u_0 = -1/2 .$$

The expected value of x_1^2 is then equal to 1/4 and this is the value of the performance measure, J, for the optimal open-loop controller.

An optimal feedback controller can be implemented by first observing the value of x_0 and then assigning the closed-loop control action:

$$u_0(x_0) = -x_0 .$$

This control action always yields a terminal value of x_1 equal to zero and an expected value of zero for the performance measure. The closed-loop control policy is thus better than the open-loop control policy in terms of the specified performance measure.

This example can be extended to include a situation in which measurement errors occur when the state x_0 is being measured. For example, suppose that there is an ϵ probability of measuring the wrong state. This can be modeled by an observation process in which a random variable y_0 is obtained. This random variable takes on the true value of x_0 with a probability of $(1-\epsilon)$ and the wrong value of x_0 with a probability of ϵ.

In this extended case the best control policy is a closed-loop control policy given by:

$$u_0 = -E\{x_0 \mid y_0\} .$$

This expression implies that the control input u_0 takes on the negative of the expected value of x_0 given the observed measurement value y_0. Since:

$$E\{x_0 \mid y_0 = 1\} = (1-\epsilon) \text{ and}$$

$$E\{x_0 \mid y_0 = 0\} = \epsilon ,$$

the performance measure J takes on a numerical value equal to $\epsilon(1-\epsilon)$. The previous closed-loop control case corresponds to the case when ϵ equals zero, that is the state x_0 is measured by y_0 known without error, and the previous open-loop case corresponds to a value of ϵ equal to 0.5, that is when the measurement y_0 effectively contains no information about the state x_0.

This example illustrates a basic distinction between stochastic and deterministic control problems. In a stochastic optimal control problem the appropriate control action at any point in time during the dynamic system's evolution may be based on a noisy observation of the dynamic system state. The optimum value of the performance measure depends on the quality of these observations, measured in the above example by the probability ϵ, and the constraints on the set of control inputs.

In many stochastic systems the observation process permits an intermediate step, the computation of the conditional probability distribution of the state given the value of the observation. This intermediate step is called filtering. Filtering is an essential ingredient of nearly all stochastic control problems, and one particular filter, the Kalman filter, was presented in Chapter 6.

11.4 Discrete-Time Stochastic Optimal Control

When computers are applied to control a dynamic system, a discrete-time mathematical model of the underlying continuous-time process must be developed. The design of the control system is then based on this mathematical model. This discrete-time representation is normally developed by integrating the continuous-time state transition equations of motion over one sample time to obtain a set of discrete-time state transition equations, and modeling the noise input and measurement errors by sequences of random variables.

The discrete-time dynamic system model which results from this process is a nonlinear time-varying difference equation having the following form and illustrated in Figure 11-4:

$$x(k+1) = f\big(x(k),\ u(k),\ w(k),\ k\big),\ k = 0,\ 1,\ 2,\ \ldots,\ K .$$

$$y(k) = g\big(x(k),\ z(k),\ k\big) ,$$

where

$x(k)$ = the system state at time k
$u(k)$ = the applied control input at time k
$w(k)$ = a noise or disturbance input at time k
$y(k)$ = the system output at time k
$z(k)$ = a measurement error at time k

The state transition equation describes the manner in which the n state variables which comprise the vector $x(k)$ evolve over time in response to an applied m-dimensional control input $u(k)$. The p-dimensional observable output $y(k)$ depends on the state variables $x(k)$ and typically on a set of p random variables which model the measurement errors.

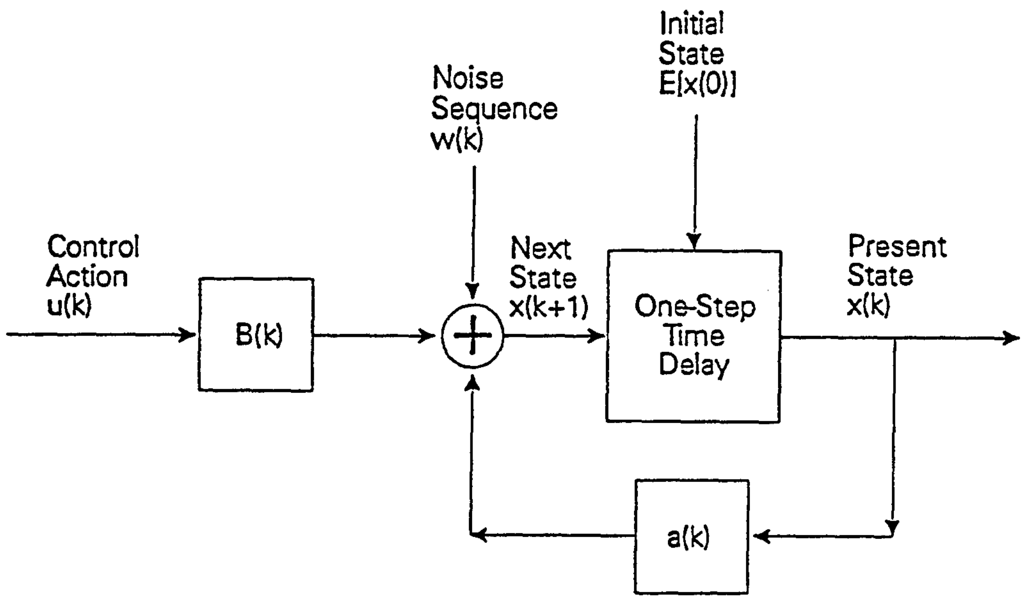


Figure 11-4. Discrete-time stochastic process that models noise input and measurement errors as a sequence of random variables.

The functions f(.) and g(.) are assumed to be specified. The random variables $w(k)$ and $z(k)$ are usually assumed to represent independent noise sequences. Often, $w(k)$ is a white noise process. Then the sequence of states $x(k)$ is called a Markov process. The applied control input may be open loop, of the form $u(k)$, $k = 0, 1, 2, \ldots, K$, or closed-loop, of the form $u(x(k))$, $k = 0, 1, 2, \ldots, K$. The probability density functions of $w(k)$ and $z(k)$ must be specified, as well as that of the initial dynamic system state, $x(0)$.

When applied to this dynamic system optimal stochastic control involves the optimization of a performance measure, J, which usually consists of the sum of a number of terms:

$$J = \sum_{k=0}^{k=K} h\left(x(k),\, u(k),\, k\right) .$$

The function h(.) is a mathematical definition of the cost of operating the dynamic system over a single time step, starting at time k in a state x(k) and applying a control action u(k). If a reward is indicated rather than a cost, this performance measure must be maximized subject to the constraints implied by the dynamic system's state transition and output equations and any additional constraints which restrict the selection of a control action. The time of control may be finite and equal to K, or infinite. For an infinite time of control the indicated performance measure may take on an infinite value for any control sequence. For that reason an exponential weighting factor, also called a discount factor, may be included to ensure convergence of the performance measure sum:

$$J = \sum_{k=0}^{k=K} a^k h\left(x(k),\ u(k),\ k\right) .$$

The presence of the random variables w(k) and z(k) means that present values of the state x(k) and future values of the state x(k+1) will always be uncertain in the sense that they cannot be computed or predicted exactly in advance. For this same reason, the performance measure as stated above is also a random variable whose numerical value varies from trial to trial as the performance of the dynamic system and its controller are tested.

Rather than using the indicated performance measure, a more suitable performance measure is the expected value:

$$J = E\left\{\sum_{k=0}^{k=K} a^k h\left(x(k),\ u(k),\ k\right)\right\} .$$

This expected value must be computed over all possible combinations of state variables, control inputs, and noise sequences.

If the state of the system is observable, then the control action can be implemented by a feedback controller and the applied control action can be written as:

$$u(k) = u\left(x(k),\ k\right) = ,\ 1,\ 2,\ \ldots,\ K .$$

The sequence of functions u = {u(0), u(1), ...} is formally called a control policy. A control policy specifies in detail how to compute the control input u(k) at each stage k of a stochastic control problem. The basic method for determining optimal control policies in many problems of interest is the computational method of dynamic programming. In applying the dynamic programming algorithm, the expected cost of operation is computed over all combinations of state and control variables for each stage of operation.

If the state of the dynamic system is not completely observable, the values of the applied control inputs and the observed outputs can be recorded and this data can be used to recursively estimate the state variable vector x(k). This process is the subject of estimation theory, discussed in Chapter 6 of this report. If the precise nature of the dynamic system is unknown the methods of system identification, discussed in Chapter 5, can be used to derive a best-fit mathematical model of the observed data.

A strictly analytical approach toward solving the general discrete-time stochastic optimal control problem does not usually lead to a closed-form solution, except in the linear quadratic Gaussian control problem discussed below. The analytic approach does yield the important conclusion that an optimal stochastic controller must perform two distinct functions. First, at each instant of time, the controller must update its estimate of the dynamic system state. This is done by evaluating the conditional probability density function for the state x(k), given the observation y(k). Second, the optimal controller must use its estimate of the system state x(k) to determine an appropriate feedback control input u(k). The optimal controller for a stochastic system is thus a feedback controller. The general structure of this controller is illustrated in Figure 11-5.

As mentioned, the complete solution for the optimal stochastic control problem can only be found analytically for a few cases, one particular case being when the state transition equations are linear, the performance measure is a quadratic function of the state and control variables, and both the noise models and the probability density function of the initial state are described by Gaussian random variables. The form of the solution for this class of problems, called the linear quadratic Gaussian problem, is well known. The solution to this problem requires the implementation of a feedback controller identical to that in the deterministic linear quadratic control problem, and the implementation of a Kalman filter to provide an estimate of the dynamic system state. The feedback controller determines the optimal control action based on the estimated values of the system state variables. The structure of an optimal controller for the linear quadratic control problem is shown in Figure 11-5.

Much work has been done to investigate and apply the solution of the linear quadratic Gaussian control problem to a variety of practical applications. More will be said about the application of these results later in this report. For nearly all other optimal stochastic control problems, the functions of state estimation or control computation must be done in a suboptimal manner.

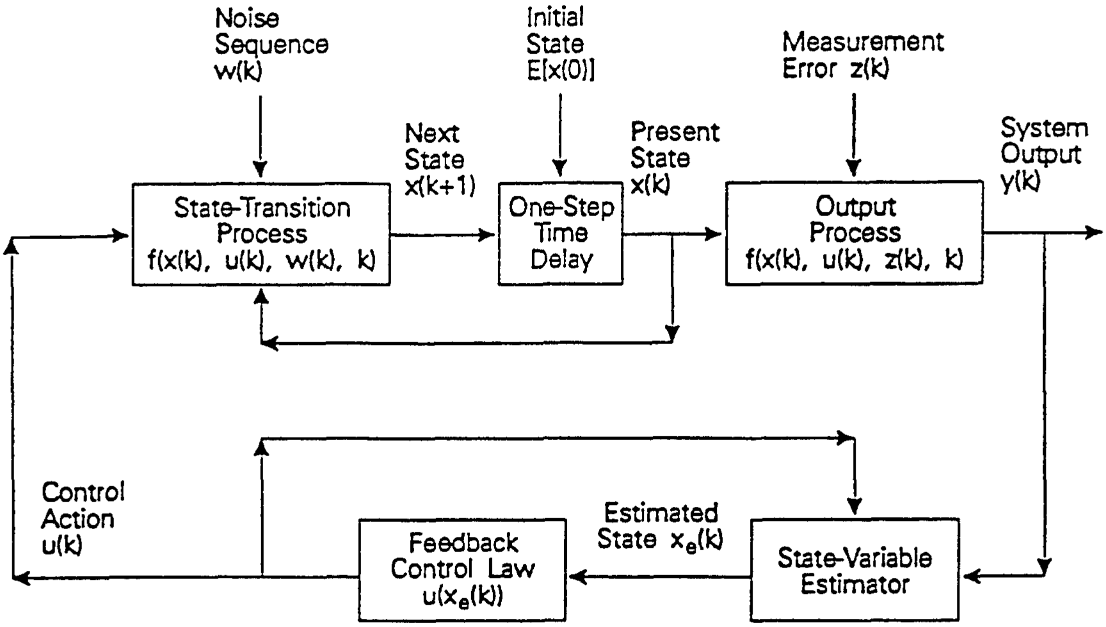


Figure 11-5. Optimal stochastic controller.

Most stochastic control problems encountered in practice are not of the linear quadratic Gaussian form, and most control problems of interest involve some nonlinearities or time delays which are not accounted for in the linear quadratic Gaussian formulation. Performance measures other than a quadratic form of the state and control variables are often of more interest. For example, a minimum time of transfer from an initial state to the origin may be more desirable than a transfer from an initial state to a state near the origin over an undetermined time of control. The noise and disturbances may not be Gaussian in nature. For these reasons, practical controllers are suboptimal in design.

11.5 Linear Quadratic Gaussian Control Problem

The linear quadratic Gaussian control problem is one particular stochastic optimal control problem for which analytical results are available[11.2].

For nearly all other stochastic optimal control problems, solutions must be determined in numerical form. Results regarding the solution of the linear quadratic Gaussian control problem are available for both continuous and discrete-time linear systems. In the following discussion the discrete-time case is emphasized since this form must be used for digital computer implementations. The dynamic system is modeled by a linear time-invariant difference equation of the form:

$$x(k+1) = Ax(k) + Bu(k) + Cw(k)$$

where A, B, and C are constant matrices, w(k) is a white noise sequence, and the initial state of the system x(0) is modeled by a multivariate normal probability density function having a mean value vector, m, and a covariance matrix, P.

The performance measure for the linear quadratic Gaussian control problem is the expected value of a quadratic form:

$$J = E\left\{\sum_{k=0}^{k=K-1}\left[x^T(k)\,Q\,x(k) + u^T(k)\,R\,u(k)\right] + x^T(K)\,F\,x(K)\right\} .$$

The constant matrices Q and F must be non-negative definite matrices and the constant matrix R must be positive definite.

The solution to this problem is the optimal control policy given in feedback form by:

$$u(k) = -M(k)\,x(k),\ k = 0,\ 1,\ 2,\ \ldots,\ K-1$$

where the gain matrix M(k) is given by the following matrix equation:

$$M(K) = \left[B^T S(K-k-1)\,B + R\right]^{-1} B^T S(K-k-1)\,A,\ k = 0,\ 1,\ 2,\ \ldots,\ K - 1$$

and the matrix S(k) is the solution to the matrix Ricatti equation:

$$S(k+1) = A^T S(k)\,A + Q - A^T S(k)\,B\left[B^T S(k)\,B + R\right]^{-1} B^T S(k)\,A\ ,$$
$$S(0) = F\ .$$

The optimal control policy is a linear feedback of the state x(k) at each time k. One important observation is that the optimal control action does not, in this case, depend on the noise matrix C. The feedback gain matrix can thus be pre-computed and stored for later use.

11.6 The Separation Principle

The linear quadratic Gaussian control problem has as its solution a feedback controller which depends on the availability of information regarding the state of the system x(k) at each time k. If the observations are noisy the dynamic description of the process must be modified. The noisy observations can be modeled by a discrete-time algebraic equation of the form:

$$y(k) = H\,x(k) + v(k)\ .$$

Here y(k) represents an observed linear combination of the state variables x(k) obtained at time k and v(k) is a white noise sequence which is statistically independent of the white noise sequence w(k). The matrix H is assumed to be constant.

The best estimate of the state of the system at time k is an expected value:

$$x'(k \mid k-1) = E\{x(k) \mid y(1),\ y(2),\ \ldots,\ y(k-1)\}\ .$$

This estimate can be provided by a Kalman filter, and the optimal control policy for the case of noisy observations is given by:

$$u(k) = -M(k)\,x'(k \mid k-1)\ .$$

The applied control input at time k is thus a linear combination of the estimated state of the dynamic system, x' rather than a function of the true but unobservable state x. The sequence of control gains may be pre-computed as before.

This result is called the separation principle. The separation principle states that, for the special case of the linear quadratic Gaussian stochastic optimal control problem, the optimal control action is determined based on the estimated value of the true state x(k) and the operations of state estimation and feedback control can be separated and designed independently. The Kalman filter, a means for estimating the values of the state variables of a linear dynamic system presented in Chapter 6 of this report, performs the task of computing the conditional probability density function of the dynamic system state based on the observed information. The expected value of the state, based on the observations of y(k), is then used as the input to an optimal feedback controller which generates the applied control action. The optimal feedback controller is designed using the method outlined in Chapter 3 of this report. That method determines a deterministic optimal feedback controller for the linear quadratic optimal control problem.

11.7 Design of Suboptimal Stochastic Controllers

According to the separation principle, the two required functions of parameter estimation and closed-loop system control can be implemented in two distinct algorithms. These algorithms can then be cascaded to produce an optimal control system in the linear quadratic Gaussian case and a suboptimal control system in most other cases. The resulting suboptimal controller is likely to be very nearly optimal, compared to suboptimal controllers designed by other methods.

A direct approach to designing a suboptimal stochastic controller is to assume that the control problem is a linear quadratic Gaussian problem. One way in which this can be done is to select a reference trajectory and then linearize the state transition equations about the reference trajectory to obtain a set of linear state transition equations. The extended Kalman filter, based on a linearized version of the dynamic system's state transition equations, can be used as a suboptimal state estimator when the state transition equations are nonlinear. Then a linear feedback controller can be designed based on the methods outlined above. As a starting point in this process, a linear, time-invariant model of the dynamic system must be developed. The linear model can either be in the form of a transfer function based on a z-transform analysis, or in the form of a difference equation with constant coefficients.

As an example, consider a continuous-time dynamic system described by the following state transition and output equations:

$$\frac{dx(t)}{dt} = -10\,x(t) + u(t) + w_c(t),\; x(0) = 1.0 \;,$$

$$y(t) = x(t) + v_c(t) \;.$$

The input noise process $w_c(t)$ is white, Gaussian, zero-mean, with a covariance of $Q_{wc} = 0.001$. The measurement noise process $v_c(t)$ is white, Gaussian, zero-mean, with a covariance of $R_{vc} = 0.001$.

The performance measure to be minimized is:

$$J = E\left\{ x^T(T)\,F\,x(T) + \int_{\tau=0}^{\tau=T} \left[x^T(t)\,Q\,x(t) + u^T(t)\,R\,u(t) \right] d\tau \right\}$$

with F = 1/2, Q = 1/4, R = 1/2, and T = 0.40 seconds. This is a linear quadratic Gaussian stochastic optimal control problem. The optimal control policy for this system is, by application of the separation principle,

$$u^*(t) = -R^{-1} B^T W(t)\, x_e(t) \;,$$

where W(t) is the time-varying solution of the matrix Ricatti equation:

$$\frac{dW(t)}{dt} = -A^T W(t) - W(t)A - Q + W^T(t) B R^{-1} W(t) \;,$$

with

$$W(T) = F,\; A = -10,\; \text{and}\; B = +1 \;.$$

An approximate solution for W(t) can be obtained by integrating in reverse time by means of simple rectangular integration with a final time T and a time-step of δt:

$$K = \frac{T}{\delta t}$$

$$W(K) = F$$

$$W(k-1) = W(k) - \delta t * \left[-A^T W(k) - W(k)A - Q + W^T(k) B R^{-1} B W(k)\right],$$

$$k = K, K-1, \ldots, 2, 1.$$

The resulting values of the matrix W(k) must be stored for later use.

The state transition equation and output equations can be similarly converted to discrete-time form:

$$x(k+1) = \left(1.0 + (-10)\right)x(k) + \delta t * (+1) * u(k) + w_d(k),$$

$$y(k) = x(k) + v_d(k),$$

where $w_d(k)$ is a white, Gaussian noise sequence with zero mean value and covariance $Q_{wd} = Q_{wc}\delta t$, and $v_d(k)$ is a white, Gaussian noise sequence with zero mean value and covariance $R_{vd} = R_{vc}$. The sample time δt was selected as 0.004 seconds so $K = 100$. The discrete-time state transition and output equations become:

$$x(k+1) = 0.96\,x(k) + 0.004\,u(k) + w_d(k),\ x(0) = 1.0,$$

$$y(k) = x(k) + v_d(k).$$

The Kalman filter algorithm for a discrete-time dynamic system described by the following state transition and output equations:

$$x(k+1) = A(k)x(k) + B(k)u(k) + w(k),$$

$$y(k) = C(k)x(k) + D(k)u(k) + v(k),$$

has been discussed elsewhere in this report and is summarized here for reference:

(0) Set $k = 0$.
Input the matrices A, B, C, D, Q_w, R_v, G(0) and $x_e(0)$.

(1) Compute the matrices $P(k) = R + CG(k)C^T$ and $P^{-1}(k)$.

(2) Compute the matrix $M(k) = AG(k)C^TP^{-1}(k)$.

(3) Compute the state estimate:
$x_e(k+1) = Ax_e(k) + M(k)[y(k) - Cx_e(k) - Du(k)] + Bu(k)$.

(4) Compute $G(k+1) = [A - M(k)C]G(k)A + Q$.

(5) Set $k = k + 1$.
Go to Step (1).

To apply this Kalman filter algorithm to the example at hand set A = 0.96, B = 0.004, C = +1.0, D = 0.0, Q_w = (0.001)(0.004), and R_v = 0.001. An initial estimate of $x_e(0)$ = 0.0 and an initial estimate of G(0) = 1.0 will be used.

Figure 11-6 shows the resulting state variable trajectories. Note that the solid line indicates the true, but directly unobservable state x(t), while the dotted line indicates $x_e(t)$, the estimate of x(t) provided by the Kalman filter. After a brief initial transient, the Kalman filter provides an estimate of the state which minimizes the mean square error.

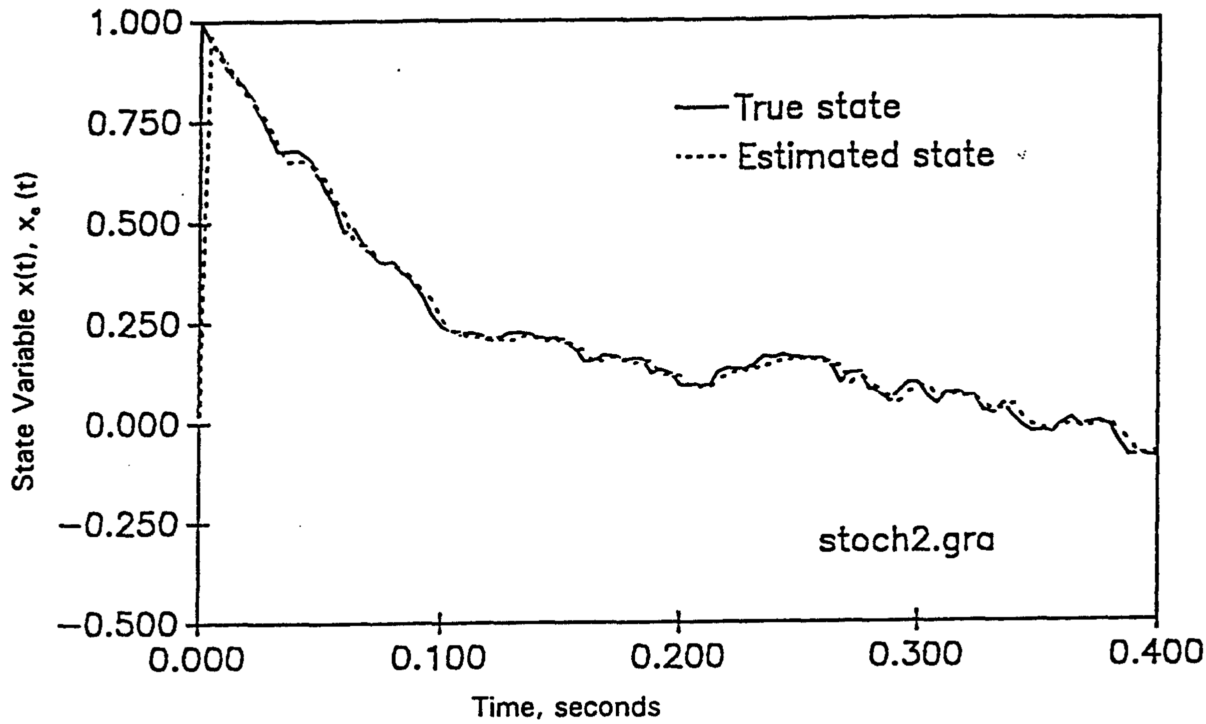


Figure 11-6. Linear-quadratic-Gaussian state variable trajectories.

The optimal control action computed as a feedback of the estimated state is plotted in Figure 11-7. After a brief initial transient, the feedback control remains relatively small as the state of the system decays naturally over time despite the presence of the noise input disturbance. As the time available for control decreases, the time-varying gain increases as indicated in Figure 11-8. This causes the control action to also increase rapidly as the time available for control approaches zero. This result is typical of linear quadratic Gaussian controllers.

The nature of the noise and measurement error processes has a significant effect on the state variable trajectories and feedback control actions in a linear-quadratic-Gaussian control problem.

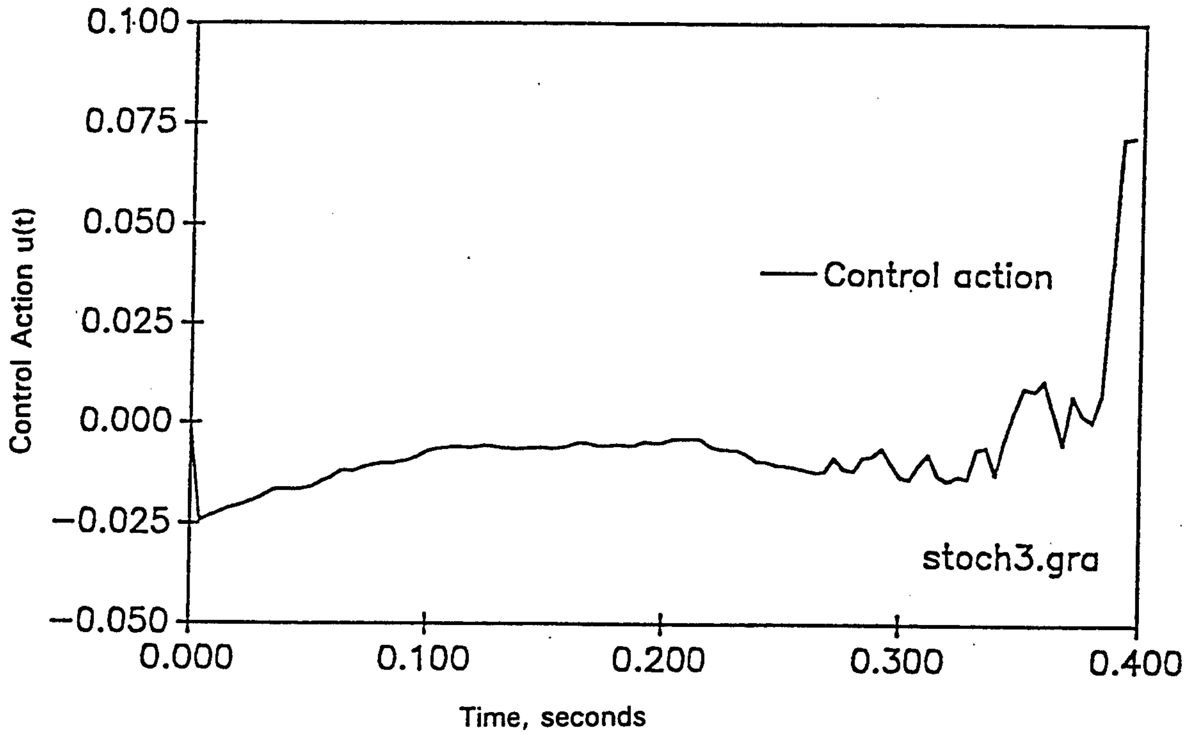


Figure 11-7. Linear-quadratic-Gaussian optimal control action.

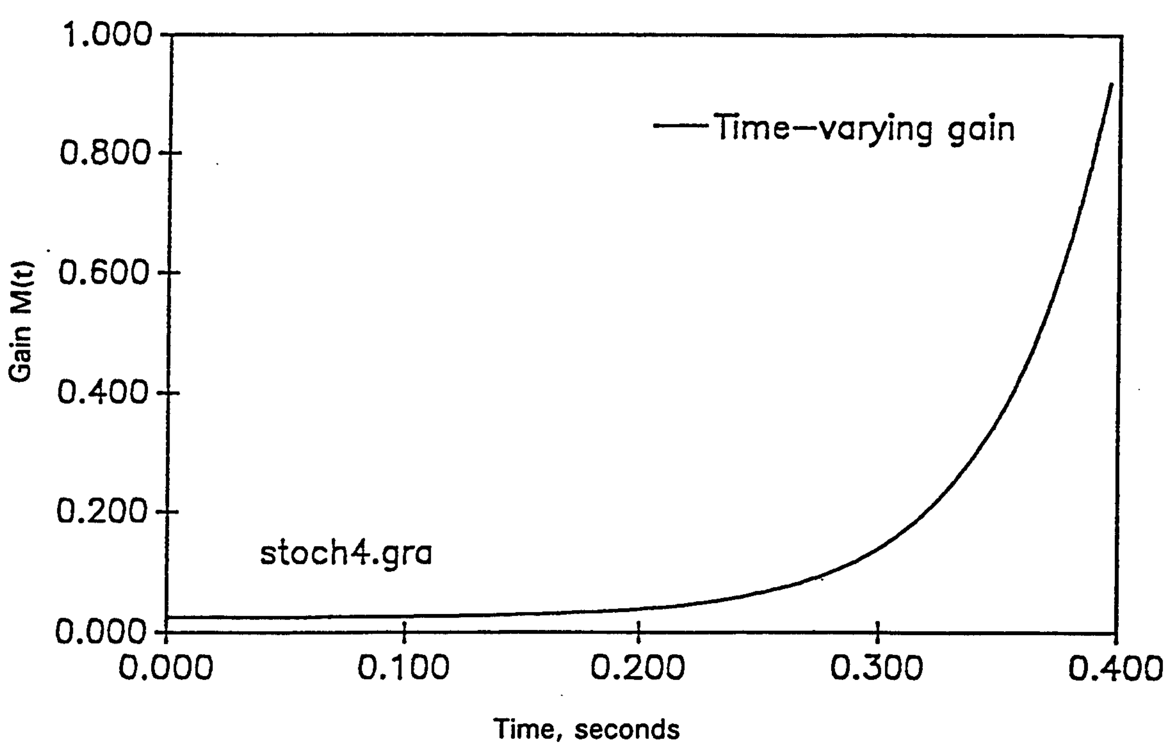


Figure 11-8. Time-varying gain resulting from linear-quadratic-Gaussian control.

Figures 11-9 and 11-10 show one sample of the state and control trajectories for the same dynamic system as above but with the input noise and measurement error covariances increased to 0.1. Compare Figures 11-6 and 11-7 to Figures 11-9 and 11-10, respectively. Note that the motion of the state x(t) indicated by the solid line appears much more erratic. After a brief initial transient, the Kalman filter again provides a good estimate of the state trajectory, indicated by the dotted line. The resulting control action is plotted in Figure 11-10. The magnitude of this control action is larger, a result of the larger variations in the estimated state trajectory.

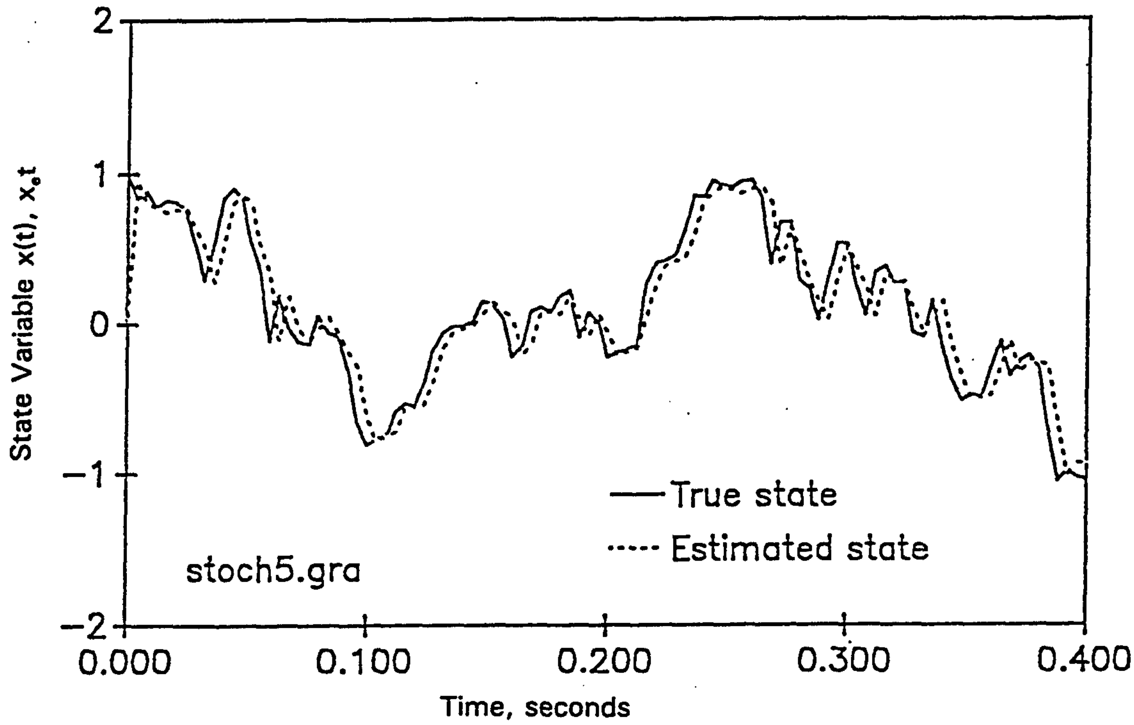


Figure 11-9. Effect of increased noise on linear-quadratic-Gaussian state variable trajectories.

State trajectories for five trials are plotted in Figure 11-11 to provide an indication of the performance of the stochastic optimal controller when the initial state of the dynamic system, x(0), is a random variable and the input noise and measurement covariances were both initialized at 0.001. Note that in all five trials the state decays naturally toward the zero level, but the overall motion is dramatically affected by the noise process. Recall that the optimal stochastic controller minimizes an expected value. This distinction does not guarantee the minimization or attainment of any final value or particular trajectory.

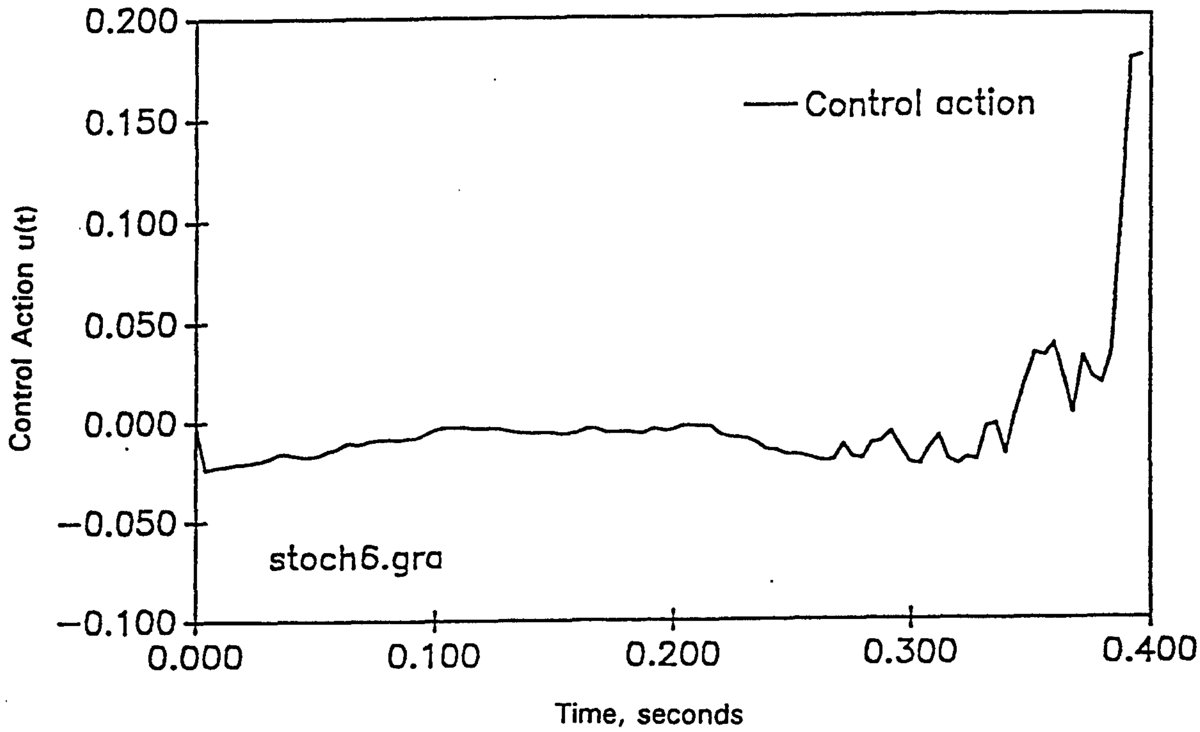


Figure 11-10. Effect of increased noise on linear-quadratic-Gaussian control action.

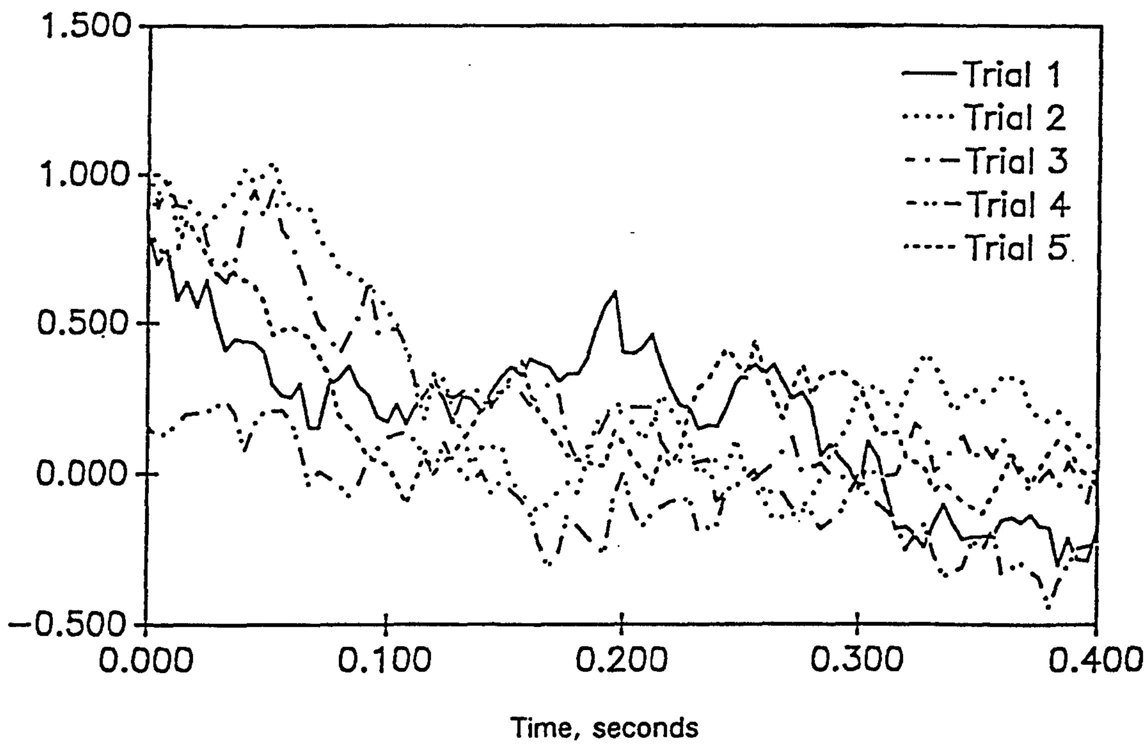


Figure 11-11. Effect of process state trajectories on the performance of a stochastic optimal controller.

11.8 Extremal Control Systems

Most controllers, whether optimal, classical, stochastic, or adaptive are implicit with respect to the cost functions and performance measures upon which their designs are based. This means that the controller does not require an explicit measurement of the cost function in order to compute the applied control input. The cost functions and performance measures indirectly affect the control action in that the control law is initially designed with the performance measure in mind.

There is an alternative approach in which the cost functions and performance measure are directly measured and this measurement is in turn used as the input to a feedback controller which applies a control action u which minimizes the performance measure. A controller of this type is designed to search for an extreme or minimum value in the mathematical relationship between the control input u and the performance measure. This approach is called extremal control.

The structure of an extremal controller differs from that of a general optimal controller in several ways. The extremal controller makes no attempt to estimate any of the states of the dynamic system or any of the coefficients of a linearized dynamic system model. The extremal controller approach concentrates on the relationship between the applied control input u and the performance measure, rather than on the mathematical relationship between the control input u and the dynamic system state x or the measured output y. This relationship is highly nonlinear in most cases of interest.

The design of an extremal controller is more difficult than a design approach based on linearization. The advantage of an extremal controller is that little information about the controlled process is required. The controller needs only that information necessary to evaluate the cost functions and the performance measure. Extremal control was originally proposed in 1950, but did not receive much attention because it could not be readily implemented using then available technology. The present availability of low-cost, high-speed digital computer hardware has generated a renewed interest in extremal control methods.

11.9 Summary

A stochastic process is a dynamic system that experiences random disturbances over time. Stochastic optimal control is required when deterministic approaches do not work. Uncertainty in the operation of the control system, such as measurement errors, manufacturing tolerances, excessive noise effects, or an imprecise mathematical model, can create problems in design of the control system. A statistical approach is necessary to predict state variables. Repeated observations of the disturbances may lead to some expectation value. In a stochastic optimal control problem, it is always better to implement a closed-loop control policy which measures the state or output of the system and then determines an appropriate control action. Feedback is of more value under these circumstances. The optimal controller for a stochastic system is a feedback controller. Discussion was presented on one of the cases where an analytical solution can be obtained for a stochastic optimal controller, the linear-quadratic-Gaussian control approach. Other solutions must be sub-optimal.

REFERENCES

11.1 Kushner, H., Introduction to Stochastic Control, Holt, Rinehard and Winston, Inc., New York, 1971

11.2 IEEE Transactions on Automatic Control Special Issue on the LQG Problem.

This page is intentionally blank.

CHAPTER 12
DIFFERENTIAL GAME THEORY

12.1 Introduction

Game theory involves the formal mathematical study and analysis of abstract models of conflict situations. The emphasis in game theory is placed on the decision-making process and, as a result, game theory has much in common with optimization theory and optimal control[12.1] discussed earlier in Chapters 9, 10, and 11.

The general notion of a game is a familiar one, and may be illustrated by a typical parlor game. Two or more players start from an initial point and proceed through a set of personal moves. Each player chooses each move from among several possibilities. The final outcome and reward, if any, depend on the combinations of strategies used by the players. The conflict modeled by the game structure may be a parlor game, a military battle, a political campaign, or competition between two firms in the same industry.

In addition to moves selected by the players, a game may involve chance or random moves resulting from the throwing of a die, the spinning of a wheel, or the shuffling of a deck of cards. The random effects of chance can be included at various levels in a game. Chess is one example of a game in which there are no chance moves beyond determining which player moves first. Each player's sequence of moves is determined primarily by skill. Bridge, by comparison, involves a considerable amount of chance as well as skill. Finally, roulette is entirely a game of chance and no skill is required to determine a player's next move, except perhaps determining when to quit.

The information available to each player is an important factor of any game. Each player in a chess game knows and can recall (not necessarily from memory) each move made by themselves and their opponents. Chess is thus a game with perfect information. On the other hand, the information available to a bridge player is imperfect, since it is generally impossible to determine all of the moves made by the opponent and by random chance. The result in a game having imperfect information is that each player, when selecting their next move, does not know the exact position or state of the game. Each move must then be selected and made allowing for the possibility that the state of the game is uncertain.

Games are generally played for some form of payoff, whether cash, a tangible or intangible reward, or personal satisfaction. The payoff depends on the progress and outcome of the game and can be considered as a mathematical function which assigns a reward to each terminal state of the game.

12.2 Discrete Game Theory

The majority of research in game theory has focused on games involving two players. Discrete game theory involves the solution of optimization problems involving two opponents, or players, having conflicting interests. The formalism of discrete game theory is the basis for more advanced game structures. The structure of a discrete game can be described mathematically by a matrix whose rows and columns represent the possible inputs, or strategies, available to each player and whose entries or elements represent the outcome, or payoff, resulting from a combination of each strategy. Player A attempts to minimize the payoff and player B attempts to maximize the payoff.

Discrete games having this structure are perfect information games because both players have access to all of the information about the game, i.e., the selections of moves or strategies, the payoffs, and the goal of the other player. Table 12-1 summarizes the information available for a simple discrete game.

TABLE 12-1. INFORMATION AVAILABLE FOR A DISCRETE GAME

PLAYER A STRATEGY	v_1	v_2
u_1	$J_{11} = 2$	$J_{12} = 7$
u_2	$J_{21} = 5$	$J_{22} = 9$

If player B, the maximizer, plays first and attempts to maximize the payoff of the game, he should select the strategy corresponding to the column with the largest minimum (column v_2), since he knows that player A, the minimizer, will choose the row having the smallest minimum (row u_1). This strategy for player B is called the *maxmin strategy*.

If player A, the minimizer, plays first and attempts to minimize the payoff of the game, he should select the strategy corresponding to the row having the smallest maximum (row u_1), since he knows that player B, the maximizer, will next choose the column having the largest maximum (column v_2). This strategy for player A is called the *minmax strategy*.

The optimal strategies for this game are thus u_1 and v_2 and it does not matter which player goes first. The payoff for this game is the value $J_{12} = 7$. The mathematical condition which represents this result is:

$$\max(v_j) \min(u_i) \{J_{ij}\} = 7 = \min(u_i) \max(V_j) \{J_{ij}\} .$$

Another way to write this result is:

$$J(u_1, v_j) \leq J(u_i, v_j) \leq J(u_i, v_2) .$$

In other games having different payoff matrices, it may turn out that it does make a difference which player goes first. To generalize the concept of a discrete game and avoid this problem a set of probabilities for each player is introduced. A numerical probability value is assigned to the choice of each strategy (the selection of a row or a column) for each player, and the probabilities are assumed to be fixed. The solution of the game is then based on a computation of the expected payoff due to each player.

This result is the *minimax principle* developed by Von Neumann and Morganstern[12.1], which states that any difference between the minmax and maxmin solutions to a simple discrete game can be resolved by the use of a random strategy for each player and the computation of the expected minmax and maxmin strategies.

12.3 Continuous Games

The discrete game example presented above involved the choice of one of two available strategies for each player. A generalization of this model allows each player to select an action from a continuous strategy, defined by the real variables u and v for players A and B. Associated with these continuous strategies is a continuous payoff function J(u,v). The optimal solution to this continuous game model is a pair of strategy values u^* and v^* which satisfy the inequality:

$$J(u^*, v) \leq J(u, v) \leq J(u, v^*) .$$

This problem can be treated as a classical optimization problem involving the two variables u and v, and necessary and sufficient mathematical conditions for an optimal solution can be obtained by taking first and second partial derivatives and equating the results to zero:

$$\frac{\partial J}{\partial u} = 0, \frac{\partial J}{\partial v} = 0 \text{ , (indicating a relative optimum),}$$

$$\frac{\partial J^2}{\partial u^2} \geq 0 \text{ , (indicating a relative minimum),}$$

$$\frac{\partial J^2}{\partial v^2} \leq 0 \text{ , (indicating a relative maximum).}$$

Any point u^*, v^* which satisfies these conditions is called a saddle point. Small variations in an optimal strategy about a saddle point offer no improvement in the payoff for either player.

12.4 Prototype Differential Games

The concept of a simple game played only once can be extended to the concept of a repeated game. Repeated games are also called sequential or multistage games. In a multistage game the game structure as specified in the information matrix may change from stage to stage or play to play. As the time between the stages approaches zero, and if the concepts of continuous games outlined above are introduced, a dynamic game called a differential game results.

The theory and technology of optimal control has been shown in Chapter 9 to apply to dynamic optimization problems in which there is only a single source of control inputs. This source is the control policy determined by the control system designer. The theory of differential games extends the application of optimal control theory by applying this technology to control problems in which there are two or more competing sources of control inputs, all of which interact to drive the dynamic system from one state to another. These various sources are called, in the language of game theory, the players. A differential game is a multiplayer dynamic optimization problem.

Games of pursuit and evasion form the prototype for a large class of problems which can be investigated and solved by the application of differential game theory. In a typical problem of this class one seeks to determine how long one opponent, the evader, will survive before being caught by the second opponent, the pursuer. In some cases the evader may escape without capture.

There are many applications of this prototype model including air-to-air combat, missile versus target maneuvers, maritime surveillance, strategic balance, economic theory, and social behavior. One especially useful application is worst-case design, in which Nature is the opponent and a designer strives to find a strategy, a set of control laws, which yields the highest payoff.

In a prototype two-player differential game, two sets of control inputs are utilized, and sometimes two sets of dynamic system equations, one set for each player, are also involved. Each set of control inputs is associated with a different player or participant in the game. The goal of one of the players is to minimize a specified performance index, while the goal of the other player is to maximize that same index. Figure 12-1 illustrates the block diagram structure of a prototype differential game.

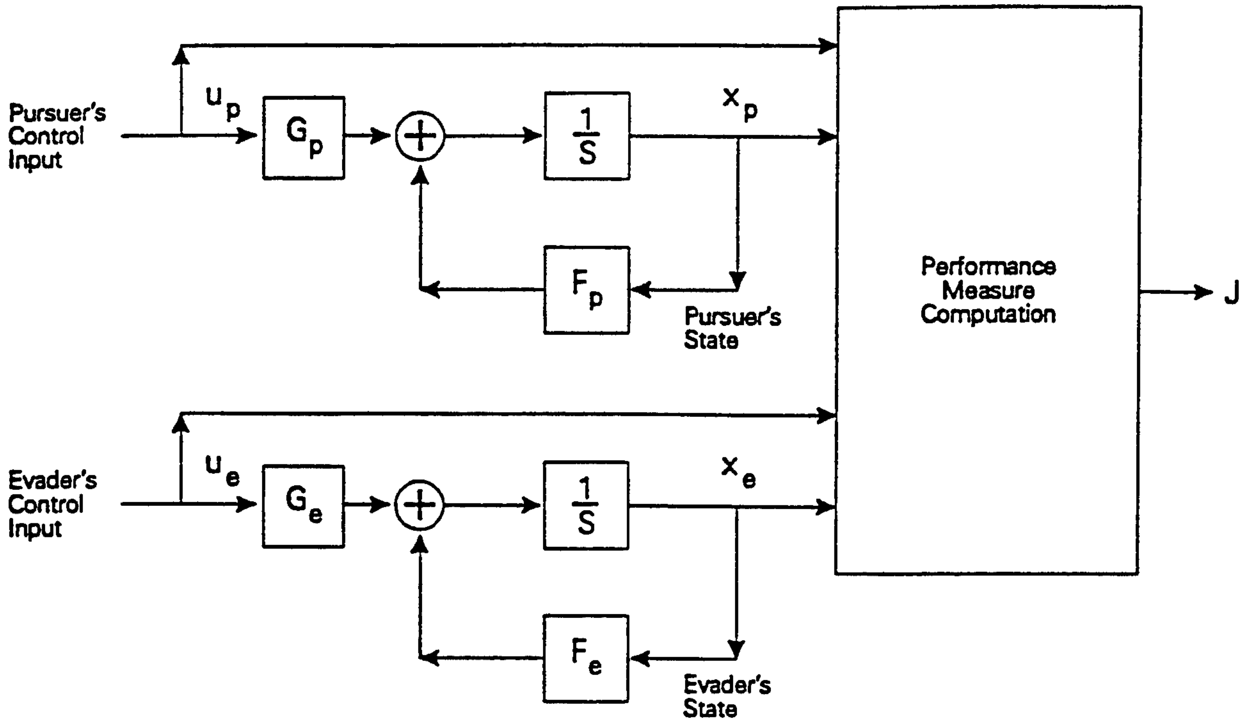


Figure 12-1. Prototype differential game.

A missile homing on a target aircraft serves as a practical example for a prototype differential game. The missile strives to be as near to the target as possible when the missile's warhead explodes. The missile bases its maneuvers on information it has obtained regarding the present and predicted position of the target. The target aircraft strives to evade the missile, maximizing the minimum distance between itself and the missile, evading the missile if possible. The target may, by means of countermeasures or maneuvers, introduce false or misleading information into the game in its efforts to avoid destruction.

12.5 The Four Elements of a Differential Game

A differential game is a mathematical generalization of a conventional multistage game in which the time interval between the game played at each stage is decreased to zero. In the limit, the sequence of moves or control actions becomes continuous. Each player must then apply a continuous control input, rather than selecting a single control action at each discrete game stage. Since the players' control actions are continuous, the state of the game is also continuous and is described by a differential equation. A differential game with two or more players has four major components.

12.5.1 State Transition Mechanism

The underlying dynamic system is specified by a state transition equation of the form:

$$\frac{dx(t)}{dt} = f\left(x(t), u_1(t), u_2(t), \ldots, u_n(t), t\right),$$

$$x(0) = x_0 ,$$

where x(t) is the system state at time t and $u_i(t)$ is the i^{th} player's control input at time t. The state of the game can be represented by an n-dimensional vector $x = (x_1, x_1, \ldots, x_n)$ of real numbers.

Each player's exercise of their control input influences the trajectory of the state variables by means of the state transition equation. The control input of each player is represented by an m-dimensional vector $u = (u_1, u_2, \ldots, u_m)$ of real numbers. The dimension of the control vector can be different for each player. The state variables x and the control variables u may be subject to sets of constraints similar to $a_i \leq u_i \leq b_i$. The numbers a_i and b_i are constants.

12.5.2 Termination Condition

The termination condition defines the end of the game. The differential game operates over time, starting at time t equal to 0, until termination is declared at some time t equal to T. Termination may occur as a result of a variety of application-dependent conditions. For example, termination may occur when the system state x(t) reaches some terminal surface. Termination may also occur when the state for player A is close enough to that of player B, or that a specified value of the system state has been reached, for example x(T) equal to 0. Termination may also occur when some predetermined maximum time of control, T equal to T_{max}, has elapsed. The termination time T may also be free but implicitly defined by the termination condition.

12.5.3 Player Performance Measures

A set of performance measures, J_i, one for each player, are defined in much the same way as the performance measure in a conventional optimal control problem:

$$J_i = K_i\left(x(T), T\right) + \int_{\tau=0}^{\tau=T} L_i\left(x(\tau), u_1(\tau), u_2(\tau), \ldots, u_n(\tau)\right) d\tau .$$

The differential game is called a zero-sum differential game if the sum of all the performance measures for all players equals zero. In a zero-sum differential game a loss for one player appears as a gain for another.

The term $K_i(x(T), T)$ is a payoff rewarded at termination. If the game begins at time t equal to 0 and terminates at time t equal to T when the state of the dynamic system is $x(T) = (x_1(T), x_2(T), \ldots, x_n(T))$, then the terminal payoff to player i is $K_i(x(T), T)$ a function which depends on the terminal state x(T) and possibly the terminal time T.

The integral term in each performance measure reflects an amount accumulated or expended over the duration of the game. This amount depends on the state and control actions at time t and possibly on t itself. Both terminal and integral payoffs can be included in a general differential game.

The solution concept for a differential game implies that if any one player uses a suboptimal strategy while all other players use an optimal strategy, that player's performance measure will take on a suboptimal value.

12.5.4 Admissible Control Strategies

A set of admissible control strategies, one for each player, must be defined. Each player's strategy indicates the extent of the information, $y_i(t)$, available to player i at time t. This information can be used to construct a control input $u_i(t)$ at time t. The information is usually available in one of the following four forms.

An open-loop control strategy permits the information available to each player to consist of only the initial state of the underlying dynamic system, or $y_i(t)$ equal to x_0. The present state x(t) is unavailable for all times other than the start of the game.

A pure feedback control strategy permits the information available to each player to consist of an exact measurement of the state of the underlying dynamic system at time t, or $y_i(t)$ equal to x(t). No memorization of the previous state is allowed.

When a memory strategy is allowed, each player may record and use information regarding the initial state x_0 as well as the history of all players' control actions. In this case $y_i(t)$ is the set of information x_0, $u_1(t)$, $u_2(t)$, ..., $u_n(t)$ for all times t from t equal to 0 to the present. The control actions are assumed to be measurable without error.

Finally, a stochastic control strategy may be permitted. In this case the information available to each player is determined by a noisy, time-varying measurement process $z_i(t) = h_i(x(t), t) + w(t)$, where w(t) is a stochastic process representing the noise or measurement errors.

This solution concept also implies that if any one player uses a suboptimal strategy while all other players use an optimal strategy, that player's performance measure will take on a suboptimal value.

12.6 Differential Games With Two Players

Much of the literature in the area of differential game theory has been devoted to discussions of the two-person, zero-sum differential game with perfect-information. Mathematically, a differential game with two players is described by a dynamic system modeled by a state transition equation of the form:

$$\frac{dx(t)}{dt} = f\big(x(t),\ u(t),\ v(t),\ t\big),\ x(0) = x_0\ .$$

In this mathematical model the system state is the vector x(t). The vectors u(t) and v(t) represent the control actions of the two competing players. The initial state is x(0). The problem usually includes various boundary conditions on the state variables x(t) and constraints on both the control and the state variables similar to those encountered in conventional optimal control problems.

Termination occurs when a set of terminal constraints g(x(T), T) are satisfied and take on values equal to 0, thus determining the end of the game at some time T. A performance measure similar to that used in a continuous time optimal control problem is constructed:

$$J = h\big(x(T),\ T\big) + \int_{\tau=0}^{\tau=T} L\big(x(\tau),\ u(\tau),\ v(\tau),\ \tau\big)\, d\tau\ .$$

The player controlling the input u(t) is assumed to minimize the performance index J, and the opponent controlling the input v(t) is assumed to maximize the same performance measure (or equivalently to minimize the negative of J). The allowable strategy in most applications is pure feedback in which both players are assumed to have perfect information and access to the precise value of the system state variables at any time t. This means that each player can record and utilize exact information about the present and past state of the dynamic system (and possibly the control inputs of both players) so as to achieve their own goal.

12.7 Numerical Solution of Two-Player Differential Games

Since both player A and player B have perfect information regarding the state of the dynamic system, which includes each other's position in the case of the missile example, a sound strategy for player A is to choose an optimal control action $u^*(t)$ which minimizes the performance measure J despite the best efforts of player B to maximize the same performance measure. This leads to a minimax control law defined implicitly by:

$$J^* = \min(u)\ \max(v) \left[h\big(x(T),\ T\big) + \int_{\tau=0}^{\tau=T} L\big(x(\tau),\ u(\tau),\ v(\tau),\ \tau\big)\, d\tau \right]\ .$$

The optimal control actions $u^*(t)$ and $v^*(t)$ also satisfy the following functional inequalities:

$$J\big(u^*,\ v\big) \le J\big(u,\ v\big) \le J\big(u,\ v^*\big)\ .$$

Necessary conditions which must be satisfied by an optimal solution to this differential game can be derived using the methods of deterministic optimal control theory. Under certain special conditions, the optimal controllers for both players can be found by introducing a set of costate variables, forming the Hamiltonian function just as in a deterministic optimal control problem, and then determining the optimal control actions $u^*(t)$ and $v^*(t)$ via the optimal value of the Hamiltonian:

$$H^* = \min(u) \max(v) H(x, u, v, p, t) .$$

The mathematical conditions under which the necessary operations can be performed are quite complicated and are not included here.

The maximum principle of Pontryagin, Section 9.3 in Chapter 9, can also be used to provide a set of necessary conditions useful for obtaining a mathematical solution to a differential game in the presence of constraints on the state and control variables.

For certain highly simplified problems, these necessary conditions can yield a solution, but a saddle point will generally not exist unless the Hamiltonian is separable into two terms, each term depending on only one control action. Further, solution of the two-point boundary value problem indicated by the necessary conditions does not automatically guarantee a saddle point solution of the differential game. These solutions may provide much information regarding the game's possible solution.

A general procedure for solving a differential game requires that, as a first step, the two-point boundary value problem be solved to yield $u^*(t)$ and $v^*(t)$. After these candidate solutions have been obtained, they must be tested for optimality according to the minmax inequality on the performance measure. This can be done by solving two one-sided optimal control problems, treating $u^*(t)$ and $v^*(t)$ as known functions in each one-sided problem, and solving for the required one-sided control input function. If the results correspond, the game solution is accepted as optimal.

The basic method for solving a differential game of the type presented involves replacing the game elements by their values and then solving a set of recurrence relations to obtain the numerical values. The recurrence relations will be seen to be differential equations. The solution process is somewhat complicated, but many of the subtleties associated with the concept of a differential game can be noted by following the solution's development. After presenting the basic equations involved in the solution of a differential game, we present one example for which a solution is available in closed form.

The value, or payoff, for a differential game which starts in the state x will be denoted by V(x). Assume that both players exercise their optimal strategies and apply optimal control inputs u^* and v^* at time t equal to 0. After a very small time increment dt, the state variables are $x + \delta x$ where the incremental change in the state x, δx, is determined by the dynamic equation:

$$\delta x = f(x, u^*, v^*)\, \delta t .$$

If the game has an integral payoff, a portion of the total payoff will have accumulated over the time increment δt:

$$\delta J = K(x)\, \delta t .$$

The game will then, in effect, restart from the state $x + \delta x$. If both players again employ their optimal strategies from the time δt on, the total payoff will be:

$$V(x) + K(x)\, \delta t + V(x+\delta x) ,$$

where $V(x+\delta x)$ is the value of the differential game starting in the new state. This term can be expanded in a Taylor series expansion about the new state:

$$V(x+\delta x) \doteq V(x) + \sum_{i=1}^{i=n} V_i(x)\, \delta x_i ,$$

where $V_i(x)$ is the change in the value of the game due to a change in x_i (a partial derivative) evaluated about the starting state x. This can also be written in terms of the incremental change in time using the state transition equation:

$$V(x+\delta x) \doteq V(x) + \sum_{i=1}^{i=n} V_i(x)\, f_i(x, u^*, v^*)\, \delta t .$$

In this equation the factor $f_i(x, u^*, v^*)$ is the state transition equation for state variable i.

The total payoff thus can be written as:

$$V(x) = K(x)\, \delta t + V(x) + \sum_{i=1}^{i=n} V_i(x)\, f_i(x, u^*, v^*)\, \delta t .$$

As the time increment δt approaches zero, this equation becomes

$$0 = K(x) + \sum_{i=1}^{i=n} V_i(x) + f_i(x, u^*, v^*) .$$

This can be written as an optimization problem in terms of the two players' control actions:

$$\min(u)\ \max(v)\left[K(x) + \sum_{i=1}^{i=n} f_i(x, u, v)\right].$$

This equation is called the main equation of the differential game. It is usually possible to interchange the order of the minimization and maximization operations when attempting to solve this equation and determine the optimal control actions for both players and the total payoff of the differential game. Note that the main equation implicitly indicates those combinations of the state x and the control actions u and v which yield an optimal solution for any starting state.

After obtaining the main equation for a given differential game it becomes possible to work backwards from the terminal surface by means of a set of differential equations. This step begins by forming the partial derivatives of the main equation with respect to each of the state variables:

$$\partial\,\frac{\left[K(x) + \sum_{i=1}^{i=n} V_i(x)\, f_i(x, u, v)\right]}{\partial x_j} = 0$$

$$\frac{\partial K(x)}{\partial x_j} + \partial\,\frac{\left[\sum_{i=1}^{i=n} V_i(x)\, f_i(x, u, v)\right]}{\partial x_j} = 0$$

$$\frac{\partial K(x)}{x_j} + \sum_{i=1}^{i=n} V_i(x)\,\frac{\partial f_i(x, u, v)}{\partial x_j} = 0\,.$$

12.8 Linear-Quadratic Pursuit-Evasion Differential Games

One class of differential games for which a closed-form solution is available is that of linear-quadratic pursuit-evasion differential games. The state transition mechanism for this class is modeled by a pair of dynamic system equations representing the state variable trajectories of the pursuer and the evader:

$$\frac{dx_p(t)}{dt} = F_p x_p(t) + G_p u(t),\ x_p(0) = x_{p0}\ ,\ \text{and}$$

$$\frac{dx_e(t)}{dt} = F_e x_e(t) + G_e v(t),\ x_e(0) = x_e 0\,.$$

The subscript p indicates the motion of the pursuer and the subscript e that of the evader. The matrices F_p, G_p, F_e and G_e are assumed to be constant during the time of the engagement. The pursuer applies the control action u(t) in an attempt to capture the evader by attaining the same state $x_p(T) = x_e(T) = x(T)$. The evader applies the control action v(t) in an attempt to evade capture and obtain a different final state than that of the pursuer, i.e., $x_p(t) \neq x_e(T)$. The final time of the engagement, T, is assumed to be fixed.

The main objective in a pursuit-evasion game is the minimization of the terminal miss distance by the pursuer and the maximization of that same distance by the evader. The miss distance can be described in a performance measure as a weighted quadratic form:

$$J = \left[x_p(T) - x_e(T)\right] A^T A \left[x_p(T) - x_e(T)\right] .$$

The weighting matrix ($A^T A$) is selected in advance to reflect the relative importance of each component of the state-variable vector x(t) at the terminal time T.

The magnitude of the control variables must be limited to reflect a practical control problem. One method for enforcing a finite control variable is to include the following integral constraints in the problem's structure:

$$\int_{\tau=0}^{\tau=T} u^T(\tau)\, R_p'\, u(\tau)\, d\tau \leq E_p ,$$

$$\int_{\tau=0}^{\tau=T} v^T(\tau)\, R_e'\, u(\tau)\, d\tau \leq E_e .$$

These constraints model the summation over time, or integral, of the control energy used by and available to the pursuer and evader. The weighting matrices R_p and R_e must be positive definite.

With limited control action available, both the pursuer and evader will intuitively apply all of their available control energy in their attempts to complete or evade capture. The inequality constraints listed above will thus hold as equalities, and may be appended to the previous performance measure:

$$J = \frac{1}{2}\left[x_p(T) - x_e(T)\right]^T A^T A\left[x_p(T) - x_e(T)\right] +$$

$$\frac{1}{2}\int_{\tau=0}^{\tau=T} \left[u^T(\tau)\, R_p u(\tau)\right] - \left[v^T(\tau)\, R_e v(\tau)\right] d\tau .$$

The weighting matrices are defined by $R_p = c_p R_p'$ and $R_e = c_e R_e'$. The positive constants c_p and c_e must be determined (by numerical solution or other means) so that the stated constraints hold as equalities. The equality constraint involving the evader's control action is subtracted from the performance measure since the evader is attempting to maximize, not minimize, the performance measure.

The solution to the linear-quadratic pursuit-evasion problem is detailed in Bryson and Ho[12.3] and will not be presented in detail here. Rather, an outline of the solution will be presented to indicate the method and results obtained. The application of these results to a problem in three-dimensional space, and a simplification resulting in conventional proportional navigation will be presented in the following sections.

A solution to the linear-quadratic pursuit-evasion problem can be developed using methods of linear system analysis and optimal control theory. First, define a set of variables $x_p'(t)$ and $x_e'(t)$ defined in terms of the pursuer's and evader's state transition matrices:

$$x_p'(t) = \theta_p(T, t)\, x_p(t) ,$$

$$x_e'(t) = \theta_e(T, t)\, x_e(t), \text{ where}$$

$$\theta_p(T, t) = e^{F_p(T-t)} \text{ and}$$

$$\theta_e(T, t) = e^{F_e(T-t)} .$$

Next, define a vector z(t), related to the miss distance, by:

$$z(t) = A\left(x_p'(t) - x_e'(t)\right) .$$

Then the performance measure for this differential game becomes:

$$J = \min(u)\, \max(v) \left\{ \begin{array}{c} \frac{1}{2}\, z(T)^T z(T) + \\ \frac{1}{2} \int_{\tau=0}^{\tau=T} \left[u^T(\tau)\, R_p\, u(\tau)\right] - \left[v^T(\tau)\, R_e\, v(\tau)\right] d\tau \end{array} \right\} .$$

Substituting the definitions for $x_p'(t)$ and $x_e'(t)$ into that for z(t) and taking a derivative with respect to time yields:

$$\frac{dz(t)}{dt} = P(t)\, u(t) - E(t)\, v(t) , \text{ where}$$

$$P(t) = A\Theta_p(T, t)\, G_p ,$$

$$E(t) = A\Theta_e(T, t)\, G_e , \text{ and}$$

$$z(0) = A\left[\Theta_p(T, 0)\, x_p(0) - \Theta_e(T, 0) x_e(0)\right] .$$

The Hamiltonian can now be written by introducing a vector of costate variables, $\lambda(t)$:

$$H = \lambda^T(t)\left[P(t)\,u(t) - E(t)\,v(t)\right] + \frac{1}{2}\left(\left[u^T(t)\,R_p\,u(t)\right] - \left[V^T(t)\,R_e\,v(t)\right]\right).$$

The necessary conditions for an optimal solution can now be written directly:

$$\frac{dz(t)}{dt} = P(t)\,u(t) - E(t)\,v(t)\,,$$

$$\frac{d\lambda(t)}{dt} = \frac{\partial H}{\partial z} = 0\,,\text{ or}$$

$$\lambda(t) = \text{a constant vector}\,,$$

$$\lambda^T(t) = \frac{d\left[\frac{1}{2}\,z^T(t)z(t)\right]}{dz} \quad \text{evaluated at } t = T\text{, or}$$

$$z(T) = \lambda(T) = \text{a constant}$$

$$\frac{\partial H}{\partial u} = 0,\text{ or } -R_p^{-1}P(t)\,\lambda(t) = -R_p^{-1}P(t)\,z(T)\,,\text{ and}$$

$$\frac{\partial H}{\partial v} = 0,\text{ or } -R_e^{-1}E(t)\,\lambda(t) = -R_e^{-1}E(t)\,z(T)\,.$$

The result is a two-point boundary-value problem in which the control actions u(t) and v(t) depend on z(T), the unknown terminal condition, and z(t) in turn depends on the unknown control actions u(t) and v(t). The solution can be implemented by a backward-sweep method. Define a matrix S(t):

$$\lambda(t) = S(t)\,z(t)\,,$$

and form the derivative with respect to time:

$$\frac{d\lambda(t)}{dt} = \frac{dS(t)}{dt}\,z(t) + S(t)\,\frac{dz(t)}{dt}\,.$$

Using all of the previous definitions, the result is:

$$u(t) = -R_p^{-1}P(t)\,S(t)\,z(t)\,,$$

$$v(t) = -R_e^{-1}E(t)\,S(t)\,z(t)\,,$$

$$\frac{dS(t)}{dt} = S(t)\left[P(t)\,R_p^{-1}\,P^T(t) - E(t)\,R_e^{-1}E^T(t)\right]S(t)\,,$$

with the additional boundary condition S(T) = I, the identity matrix.

The feedback solution for the optimal control actions u(t) and v(t) obtained by this process must then be verified to ensure that it constitutes a saddle point for the differential game. This process has been detailed[12.3] and the resulting solution shown to be optimal for both the pursuer and evader.

The solution to this class of differential games is important since it provides a basis for several numerical methods and provides approximate solutions for other classes whose underlying dynamic systems and performance measures can be linearized to yield a linear-quadratic form. Figure 12-2 illustrates the solution state trajectories for the two-player differential game represented by the state transition equations:

$$\frac{dx_p(t)}{dt} + -10x_p(t) + u(t),\ x_p(0) = 0.0\ ,$$

$$\frac{dx_e(t)}{dt} + -10x_e(t) + v(t),\ x_e(0) = 0.0\ ,$$

and the following performance measure:

$$J = 100\left[x_p(T) - x_e(T)\right]^2 + \int_{\tau=0}^{\tau=T} 10u^2(\tau) - 10v^2(\tau)\,d\tau\ ,$$

over the time interval from t = 0.00 to T = 0.40 seconds.

The state trajectories indicate the response over time of the two state variables $x_p(t)$ and $x_e(t)$. Figure 12-3 indicates the optimal control action, identical for both players. The solution to this game is such that both players apply the same control action and both terms in the integral portion of the performance measure then cancel. The final value of the performance measure is determined solely by the small difference between the final state of the pursuer and the evader. In this example, no attempt was made to ensure satisfaction of any finite limit on the control action available to each player. These limits can be imposed by varying the positive constants c_p and c_e. In that way a family of trajectories can be generated.

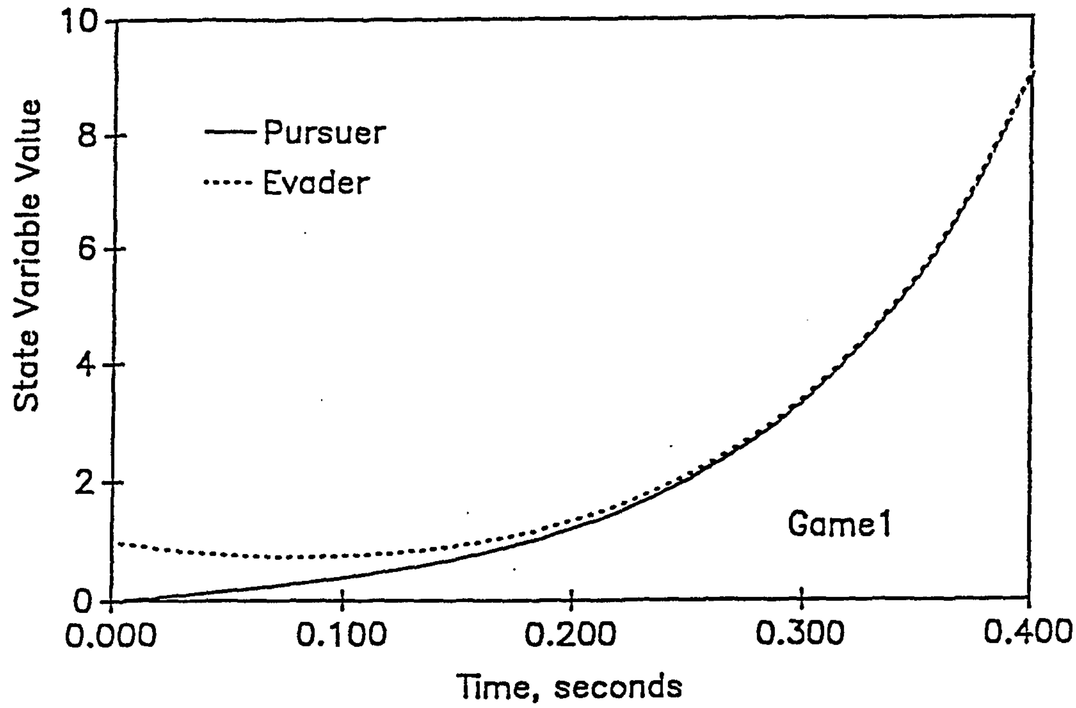


Figure 12-2. Linear-quadratic differential game trajectories.

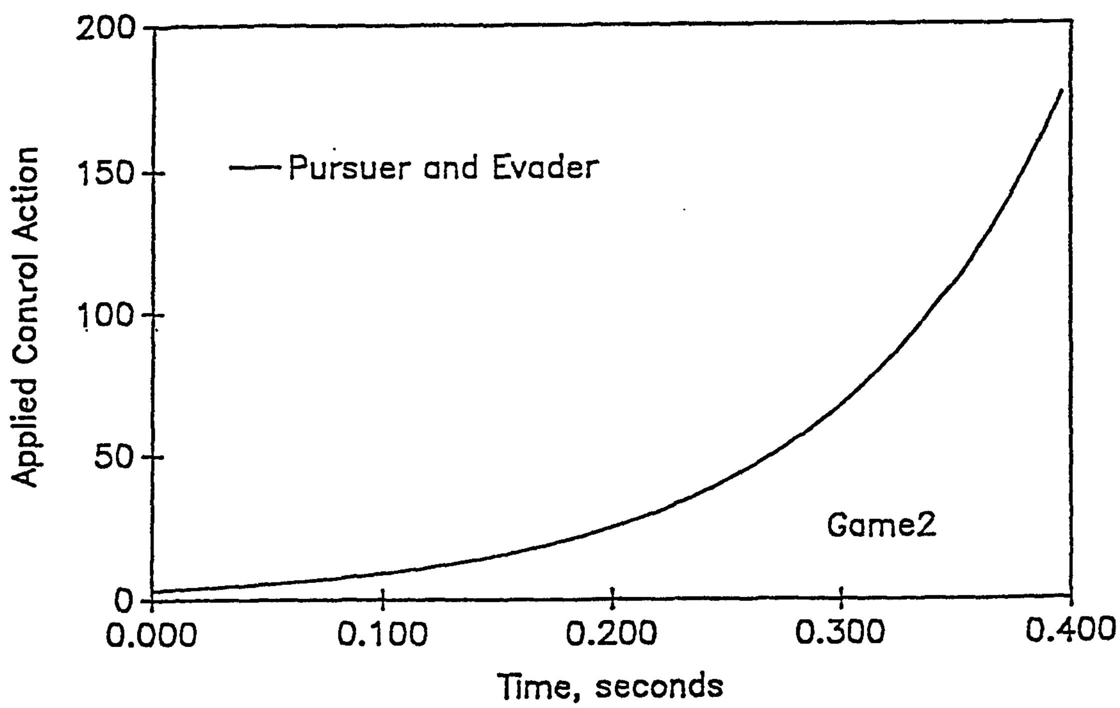


Figure 12-3. Linear-quadratic differential game optimal control actions.

12.9 Guidance Law for Three-Dimensional Target Interception

The results for the linear-quadratic pursuit-evasion differential game are applied to a point-mass model of target pursuit and interception. The equations of motion for the pursuer and evader are:

$$\frac{dx_p(t)}{dt} = v_p(t),\ x_p(0) = x_{p0}\ ,$$

$$\frac{dv_p(t)}{dt} = a_p(t),\ v_p(0) = v_{p0}\ ,$$

$$\frac{dx_e(t)}{dt} = v_e(t),\ x_e(0) = x_{e0}\ ,$$

$$\frac{dv_e(t)}{dt} = a_e(t),\ v_e(0) = v_{e0}\ ,$$

The situation is illustrated in Figure 12-4.

The positions of the pursuer and evader are $x_p(t)$ and $x_e(t)$, the velocities are $v_p(t)$ and $v_e(t)$ and the applied control accelerations are $a_p(t)$ and $a_e(t)$. In this form the motions in each direction are uncoupled, and the effects of gravity on the players' motions has been ignored. If gravity is the same for both players then a compensating term must be added to each control action.

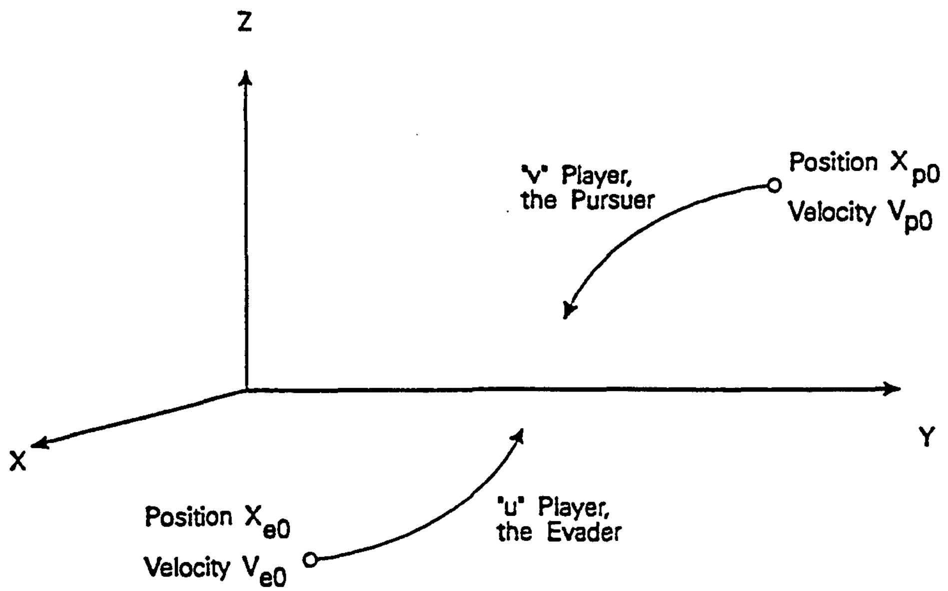


Figure 12-4. Three-dimensional target interception.

For an intercept at the specified terminal time T a suitable linear-quadratic performance measure is:

$$J = \frac{b}{2}\left[x_p(T) - x_e(T)\right]^T\left[x_p(T) - x_e(T)\right] +$$

$$\frac{1}{2}\int_{\tau=0}^{\tau=T}\left[\frac{1}{c_p}\, a_p^T(\tau)\, a_p(\tau) + \frac{1}{c_e}\, a_e^T(\tau)\, a_e(\tau)\right] d\tau \,.$$

The constants c_p and c_e relate to the energy available to the pursuer and evader. The constant b assigns a weight to the terminal miss distance. Applying the results of the previous Section, and using the fact that the equations of motion are uncoupled, the resulting optimal control actions are:

$$a_p(t) = \frac{-c_p(T-t)\left[x_p(t) - x_e(t) + \left(v_p(t)-v_e(t)\right)(T-t)\right]}{\dfrac{1}{b} + \dfrac{\left(c_p-c_e\right)(T-t)^3}{3}}$$

$$a_e(t) = \frac{c_e\, a_p(t)}{c_p} \,.$$

Note that the optimal control action is a time-varying feedback of the state variables of both the pursuer and evader, $x_p(t)$, $x_e(t)$, $v_p(t)$ and $v_e(t)$. The sign of the feedback gain for each term is determined by the denominator in the above expression. If $c_p > c_e$ then the feedback gains are always of the same sign. If $c_p \le c_e$ then the feedback gains change sign at a time t determined implicitly by:

$$\frac{1}{b} + \frac{\left(c_p-c_e\right)(T-t)^3}{3} = 0.0 \,.$$

Figure 12-5 illustrates the solution for a two-dimensional pursuit-evasion problem in the x-y plane. The initial position of the pursuer is at x_{p0} equal to (0,0) and the pursuer has zero initial velocity. The initial position of the evader is at x_{e0} equal to (2,1) and the evader has an initial x-velocity of -0.10 per second. The terminal time was set at 10.0 seconds. The parameters c_p and c_e were set at 3.0 and 2.0.

12.10 Proportional Navigation

As the parameter b increases without limit, reflecting the assignment of increased importance to the terminal outcome, interception is mathematically impossible when $c_p < c_e$. For the case when $c_p > c_e$ the optimal control actions for the pursuer and evader simplify to:

$$a_p(t) = \frac{-3\left[x_p(t) - x_e(t) + \left(v_p(t) - v_e(t)\right)(T-t)\right]}{\left[1 - \frac{c_e}{c_p}(T-t)^2\right]} ,$$

$$a_e(t) = \frac{c_e a_p(t)}{c_p} .$$

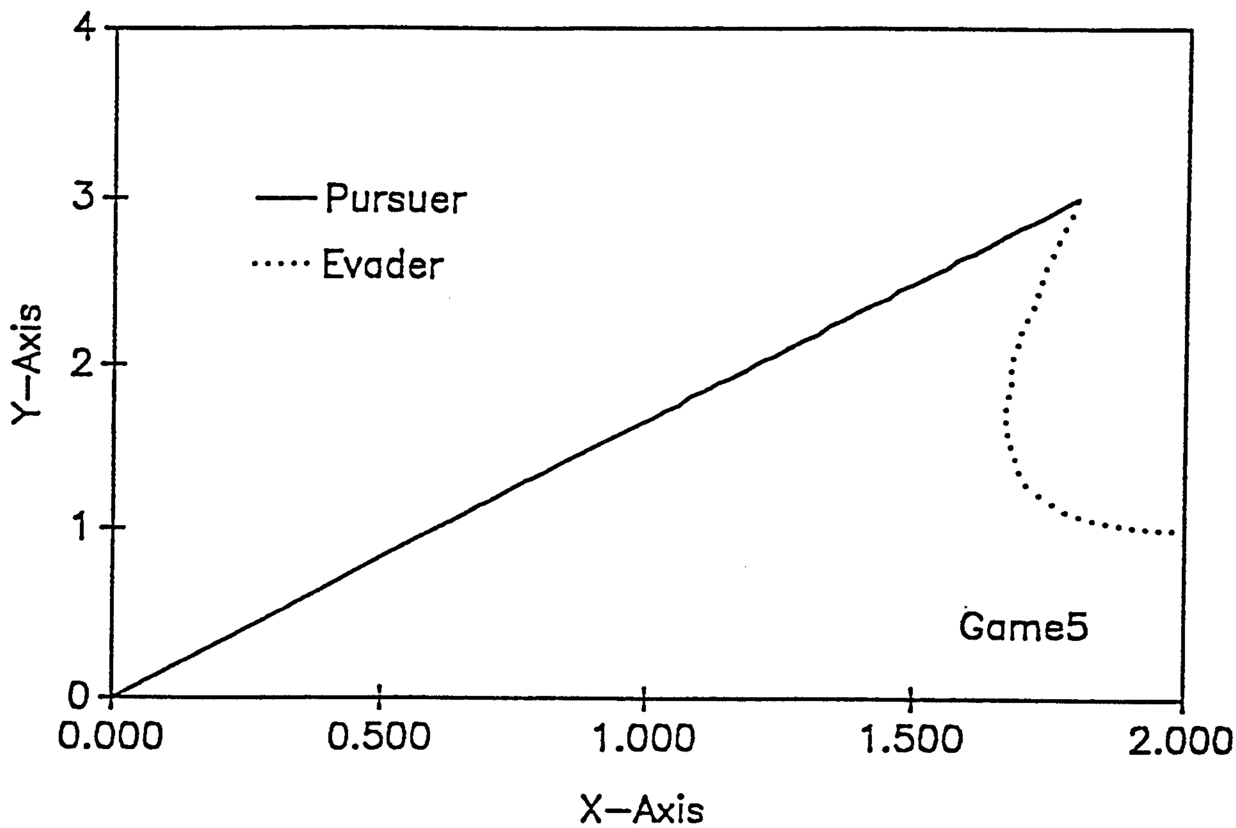


Figure 12-5. Pursuit-evasion game trajectories.

If the pursuer and evader begin the differential game at a time t on a nominal collision course at a range R and a closing velocity V = dR/dt, and if $(x_p(t)-x_e(t))$ represents a lateral deviation from the collision course, the optimal lateral acceleration to be applied by the pursuer is:

$$a_l(t) = \frac{3\,V \frac{d\sigma(t)}{dt}}{\left[1 - \frac{c_e}{c_p}\right]} .$$

where σ is the LOS angle indicated in Figure 12-6.

This lateral acceleration is simply proportional navigation with an effective navigation constant of $K_e = 3.0/(1.0 - (c_e/c_p))$. Implementation of proportional navigation for either the pursuer or the evader requires only the measurement of the line of sight (LOS) angular rate ($d\sigma(t)/dt$) measured from the pursuer to the evader. Experience has indicated that, without knowledge of the precise values of the constants c_p, c_e and the final time T in any actual engagement, an appropriate value for the effective navigation constant K_e lies between the values of 3.0 and 5.0. The lower limit pertains to an engagement involving a non-maneuverable target for which c_e equals 0.0, and the upper limit to an engagement in which the ratio c_e/c_p equals 2.5.

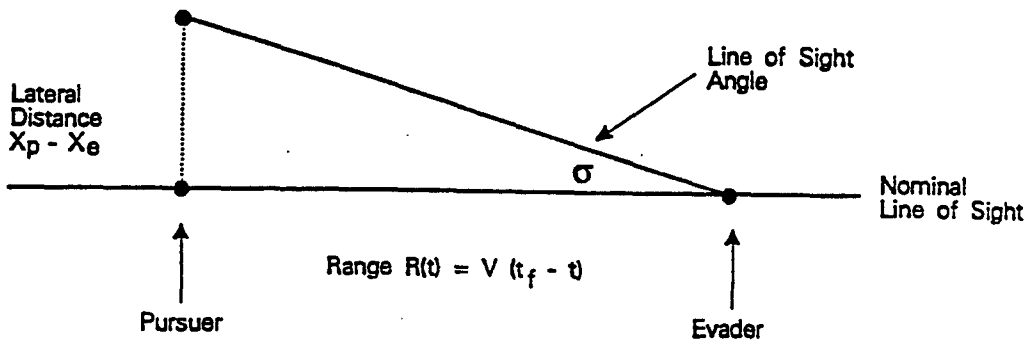


Figure 12-6. Geometry of proportional navigation.

Figure 12-7 shows the basic elements of a mathematical model of a generic seeker based on proportional navigation. The model includes limits on the various angles and angular rates attainable in a physical system[12.4], and the dynamics of each of the major blocks can be selected to closely match the physical characteristics of actual seeker hardware.

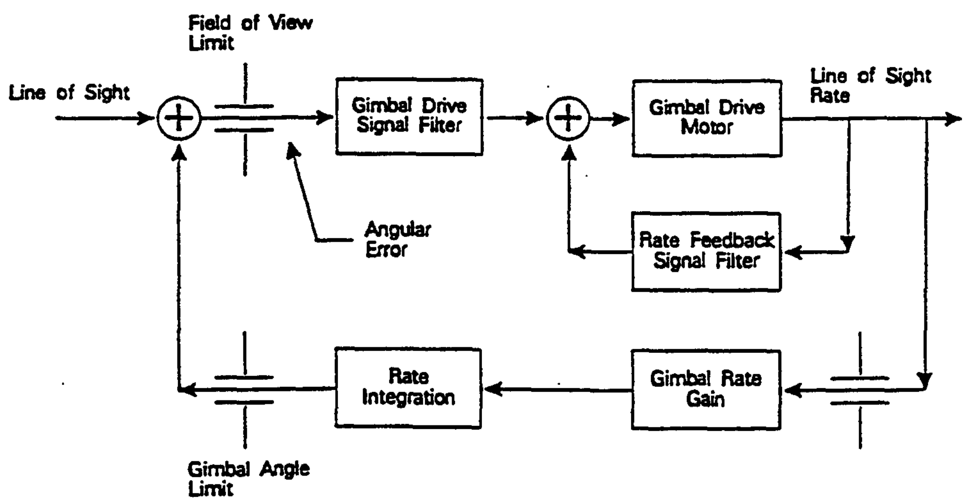


Figure 12-7. Generic seeker block diagram.

12.11 Summary

Conflict situations which require decision making are modeled by game theory. Mathematical models of games are usually based upon games where perfect information is not available. Some element of chance or probability must be introduced. Games may be discrete, continuous, or differential depending upon repetition in sequences or stages. Optimal control theory may be applied to dynamic games that involve two or more control inputs in competition. A differential game is a multiplayer dynamic optimization problem. This chapter discussed the four components of such games. Examples of linear-quadratic pursuit-evasion differential games are presented that apply to missile intercepts of targets.

For further discussion of the linear-quadratic pursuit-evasion differential game, a minmax time intercept problem with bounded control actions and several other differential game examples, the reader should consult Bryson and Ho[12.3].

REFERENCES

12.1 Owen, G., Game Theory, W.B. Saunders Company, Philadelphia, Pennsylvania, 1968.

12.2 Von Neuman, J. and Morganstern, O., Theory of Games and Economic Behavior, Princeton University Press, Princeton, New Jersey, 1947.

12.3 Bryson, A.E. and Ho, Y.C., Applied Optimal Control, John Wiley & Sons, New York, New York, 1975.

12.4 Stone, B.B. and Mokas, W.H., TRAP User's Manual, Version 2.0, Foreign Technology Division, Air Force Systems Command, Wright Patterson Air Force Base, Dayton, Ohio, 1986.

This page is intentionally blank.

CHAPTER 13
ROBUSTNESS AND SENSITIVITY

13.1 Definitions

The dynamic response of any closed-loop control system is a major design consideration. The designs of nearly all control systems are based on mathematical models which approximate the true behavior of the underlying dynamic systems. For this reason it is important to design a control system in such a way that properties of the resulting closed-loop system remain the same when the mathematical model is slightly altered. These alterations may be due to time-varying changes in the dynamic system's parameters, differences from design values due to manufacturing and assembly tolerances, or the effects of external disturbances or random variations in the system's environment. Analysis techniques which can indicate the robustness of a particular control system design are thus important control system design tools.

Figure 13-1[13.1] is a basic block diagram of a closed-loop control system for a dynamic system controlled by a digital computer. The purpose of adding the digital computer and peripheral hardware is to provide a controlled system having a satisfactory response. The notion of satisfactory response means that the dynamic system output y(t) tracks or follows the reference input r(t) despite the presence of disturbance inputs and measurement errors. The disturbance inputs to the dynamic system are indicated as w(t) in the figure, and the sensor or measurement errors as v(t).

For successful operation of the closed-loop control system it is necessary that tracking occur even if the nature or structure of the dynamic system should change slightly over the time of control. The process of maintaining the system output y(t) close to the reference input r(t), in particular when r(t) equals zero, is called regulation. A control system which maintains good regulation despite the occurrence of disturbance inputs or measurement errors is said to have good disturbance rejection. A control system which maintains good regulation despite the occurrence of changes in the dynamic system's parameters is said to have low sensitivity to these parameters. A control system having both good disturbance rejection and low sensitivity is said to be a robust control system.

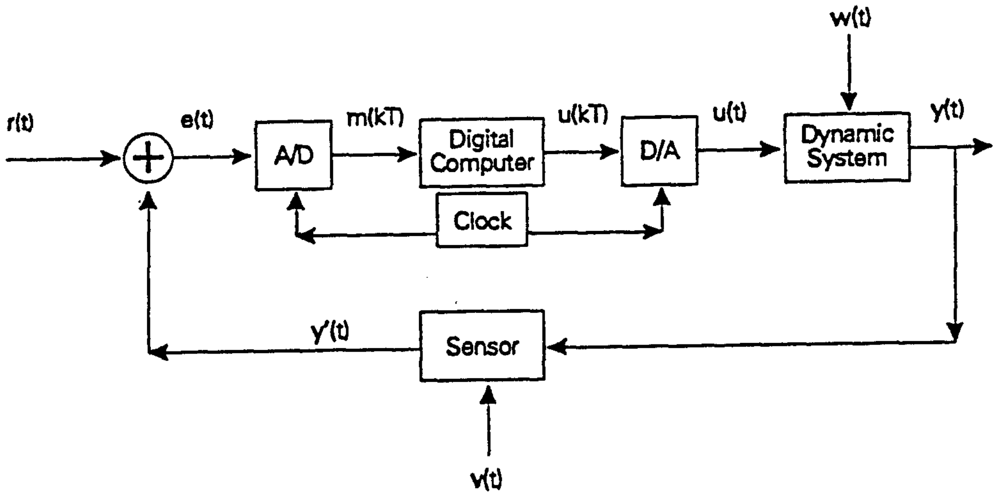


Figure 13-1. Basic control system block diagram.

13.2 The Design of Robust Control Systems

A control engineer's main design objective is the stability of the resulting closed-loop dynamic system. For many applications, the designer begins with a linear time-invariant mathematical model. This model may be derived from fundamental physical equations, frequency or transient response tests, or system identification procedures. In any case there will always be some uncertainty as to the specific numerical values of the model's parameters and indeed as to the complexity of the model itself. It is thus important that the designer produce a control system design which is not only mathematically stable but robustly stable as well.

Techniques for designing mathematically stable linear time-invariant control systems are well-known to control system engineers, and several examples of this process have been presented in Chapters 2 and 5 of this review. The design of single-input single-output control systems is commonly done using the frequency response methods of Bode or Nyquist. These techniques provide a means for determining the relative stability of a closed-loop control system by indicating, as a function of frequency, the minimum change in the model's frequency response which will cause the system to become unstable.

The gain and phase margins of a linear, time-invariant dynamic system are commonly used indicators for assessing relative stability. The gain and phase margins can also be used in a classical system design procedure to estimate the transient response of the dynamic system to certain test inputs. The gain and phase margins are often quoted as classical design specifications for single-input, single-output linear time-invariant control systems.

For multivariable control systems it is much more complex to assess the relative stability. Nyquist's method has been extended to the investigation of multivariable control systems, in which the dynamic system and its feedback controller are modeled by matrices of Laplace transfer functions[13.2].

These extensions, however, indicate only the relative stability of a proposed control system and not its robustness.

The analysis of robustness and its application to the design of multivariable feedback control systems has received considerable attention in recent years, as these systems invariably employ mathematical models for the underlying dynamic system which are time-invariant and of low order. A number of researchers have derived sufficient conditions for the stability of a dynamic system in terms of matrix expressions which yield reliable, conservative information about the robustness of a multivariable closed-loop control system.

These results are based on the initial work of Zames[13.3], who investigated the input-output stability of nonlinear dynamic systems, generalizations of the methods of Safonov[13.4], and applications of the results of Kwakernaak[13.5], Doyle[13.6], and Doyle and Stein[13.7] who studied the robustness of linear quadratic Gaussian regulators. Surveys of results concerning the design of robust controllers for multivariable systems were presented in a special issue of the IEEE Transactions on Automatic Control[13.8] dealing with linear multivariable control systems and also by Davison[13.9] and Davison and Gesing[13.10].

A robust controller provides satisfactory tracking or regulation in spite of the fact that the dynamic equations defining the underlying controlled system, or the parameters of these equations, may vary by arbitrarily large amounts. The only condition imposed is that the perturbed dynamic system resulting from a change in system parameters remains stable. The synthesis of a robust controller is thus required when the dynamic system is subject to some uncertainty.

The underlying dynamic systems in most analyses have been described by linear time-invariant state-transition and output matrix equations:

$$\frac{dx(t)}{dt} = Ax(t) + Bu(t) + Ew(t) ,$$

$$y(t) = Cx(t) + Du(t) + Fw(t) ,$$

$$y_m(t) = C_m x(t) + D_m u(t) + F_m w(t) ,$$

$$e(t) = y(t) - y_{ref}(t) ,$$

where

$x(t)$	=	the state of the dynamic system
$u(t)$	=	the control input vector
$y(t)$	=	the output vector to be controlled or regulated
$y_m(t)$	=	the outputs which can be measured
$w(t)$	=	system disturbances which can not be measured
$e(t)$	=	the system error,
	=	$y(t) - y_{ref}(t)$, and

A, B, C, D, E, F, C_m, D_m, and F_m are constant matrices whose components are assumed to be known.

The disturbance inputs w(t) can be modeled by the following state transition and linear combination equations:

$$\frac{dn_1(t)}{dt} = A_1 n_1(t) ,$$

$$w(t) = C_1 n_1(t) ,$$

and the reference inputs $y_{ref}(t)$ can similarly be modeled by:

$$\frac{dn_2(t)}{dt} = A_2 n_2(t) ,$$

$$r(t) = C_2 n_2(t) ,$$

$$y_{ref}(t) = G\, r(t) .$$

The eigenvalues of the matrices A_1 and A_2 are assumed to lie in the right hand complex plane. The dynamic systems defined by the matrix pairs (A_1, C_1) and (A_2, C_2) are assumed to be observable.

The solution of the robust controller problem for this system results in the design of a linear, time-invariant controller having inputs $y_m(t)$ and $y_{ref}(t)$ and generating a control signal u(t) such that:

(a) the resulting closed-loop control system is asymptotically stable,

(b) asymptotic tracking occurs, i.e., for any initial condition, for any disturbance in the class specified, and for any reference signal in the class specified where the error eventually approaches zero, and

(c) condition (b) holds for any arbitrary perturbations in the model of the dynamic system, resulting either from a change in the model's parameters or the dynamic equations, including a change in the model's order, which do not cause the resultant closed-loop system to become unstable.

The solution to the strong robust controller problem for this dynamic system requires the design of a feedback controller which has inputs $y_m(t)$ and $y_{ref}(t)$ and generates a control action u(t). The resulting closed-loop dynamic system satisfies the three conditions stated above and also possesses the following property.

(d) approximate error regulation occurs for any controller parameter perturbation lying in a small neighborhood about the nominal controller parameters, with the approximation becoming exact as the perturbed controller parameters approach their design values.

13.3 An Example of Robust Controller Design

A robust deterministic closed-loop control system will be designed. The resulting controller will have the ability to track, with zero tracking error, any non-decaying reference input such as a step, ramp, or sinusoid, and to reject, with zero error, a similar non-decaying input disturbance. The design procedure[13.1] explicitly includes the dynamic equations satisfied by the reference and disturbance inputs in the problem formulation. The control problem is solved in an error space (as opposed to a conventional state variable space) and the result assures that the error between the reference input and the dynamic system output approaches zero over time.

The state transition equations for the process are:

$$\frac{dx(t)}{dt} = Fx(t) + Gu(t) + G_1 w(t) \ ,$$

$$y(t) = Hx(t) + Ju(t) \ ,$$

where

x(t) = the dynamic system state
u(t) = the feedback control input
y(t) = the dynamic system output
w(t) = the input disturbance

The matrices F, G, G_1, H, and J are assumed to be constant. A pole-placement design process will be used to develop a closed-loop control system which tracks an input command or reference signal with zero steady-state error. The reference input is assumed to satisfy a second-order linear time-invariant differential equation of the form:

$$\frac{d^2 r(t)}{dt^2} = a_1 \frac{dr(t)}{dt} + a_2 r(t) \ .$$

The control system is also to reject a disturbance input. The disturbance input is also assumed to satisfy a second-order linear time-invariant differential equation of the form:

$$\frac{d^2 w(t)}{dt^2} = a_1 \frac{dw(t)}{dt} + a_2 w(t) .$$

The tracking or regulation error is defined as the difference between the reference input and the system output:

$$e(t) = y(t) - r(t) .$$

In terms of this error signal, the design problem of tracking the input r(t) and rejecting the disturbance w(t) can be considered to be the design of a controller which regulates the error signal e(t) about a reference value of zero. That is, the error should be brought to zero as quickly as possible and maintained at that value over the remaining time of control.

The desired closed-loop controller should also be robust in the sense that the regulation of the error about zero should continue despite any perturbations in the parameters of the underlying dynamic system parameters. This notion is important since in practice the mathematical state transition model defining the controlled system is never perfect and its parameters are always subject to change due to wear and tear, manufacturing tolerances, or simply uncertainty on the part of the designer.

The time-derivatives of the error signal can be formed directly:

$$\frac{de(t)}{dt} = \frac{dy(t)}{dt} - \frac{dr(t)}{dt} ,$$

$$\frac{d^2 e(t)}{dt^2} = \frac{d^2 y(t)}{dt^2} - \frac{d^2 r(t)}{dt^2} , \text{ or}$$

$$\frac{d^2 e(t)}{dt^2} = H \frac{d^2 x(t)}{dt^2} + J \frac{d^2 u(t)}{dt^2} - a_1 \frac{dr(t)}{dt} - a_2 x(t) .$$

Next, a state variable vector in error space is formed:

$$\zeta(t) = \frac{d^2 x(t)}{dt^2} - a_1 \frac{dx(t)}{dt} - a_2 x(t) ,$$

and a control variable vector in error space is similarly formed:

$$\mu(t) = \frac{d^2 u(t)}{dt^2} - a_1 \frac{du(t)}{dt} - a_2 u(t) .$$

The differential equation for the error e(t) can then be written as:

$$\frac{d^2 e(t)}{dt^2} = a_1 \frac{de(t)}{dt} - a_2 e(t) = H\zeta(t) + J\mu(t) ,$$

and the state transition equation for the error state $\zeta(t)$ then becomes:

$$\frac{d\zeta(t)}{dt} = \frac{d^3 x(t)}{dt^3} - a_1 \frac{d^2 x(t)}{dt^2} - a_2 \frac{dx(t)}{dt} , \text{ or}$$

$$\frac{d\zeta(t)}{dt} = F\zeta(t) + G\mu(t) .$$

In state variable form, an overall system state z(t) and an overall state transition equation can be written as:

$$z(t) = \left[e(t), \frac{de(t)}{dt}, \zeta(t) \right]^T ,$$

$$\frac{dz(t)}{dt} = Az(t) + B\mu(t) , \text{ where}$$

$$A = \begin{bmatrix} 0 & 1 & 0 \\ a_1 & a_2 & H \\ 0 & 0 & F \end{bmatrix} , \text{ and } B = \begin{bmatrix} 0 \\ J \\ G \end{bmatrix} .$$

If the dynamic system which now describes the overall state z(t) is completely controllable, its dynamics, in terms of pole locations, can be arbitrarily assigned. The requirement for the system described by the matrix pair (A,B) to be controllable is identical to the requirement that the underlying dynamic system described by the matrix pair (F,G) is completely controllable and does not possess a zero at the roots of the characteristic equation:

$$a_r(s) = s^2 - a_1^s - a_2 = 0 .$$

If this requirement is satisfied then a feedback control law in the form of:

$$\mu(t) = -Kz(t)$$

$$= -[K_2\ K_1\ K_0] \left[e(t)\ \frac{de(t)}{dt}\ \zeta(t) \right]^T$$

can be used to provide arbitrary dynamics for a dynamic system producing the error state z(t). Note that K_0 is a vector of gains K_{0i} corresponding to the elements of the vector $\zeta(t)$. In terms of the underlying dynamic system state x(t) and desired control input u(t), this feedback control law can be written as:

$$\left(u(t) + K_0 x(t)\right)'' = \begin{bmatrix} a_1\left(u(t) + K_0 x(t)\right)' \\ +a_2\left(u(t) + K_0 x(t)\right) \\ -K_1 e(t)' - K_2 e(t) \end{bmatrix} ,$$

where the primes represent time-derivatives.

As an illustration of the application of this approach, consider a dynamic system defined by the following second-order state transition and output equations:

$$\frac{dx_1(t)}{dt} = x_2(t) ,$$

$$\frac{dx_2(t)}{dt} = -x_2(t) + u(t) ,$$

$$y(t) = x_1(t) .$$

The system is to follow exactly a sinusoid having a frequency of ω_0 radians per second. There is no input disturbance or measurement noise present in this example. The required matrices are:

$$F = \begin{bmatrix} 0 & 1 \\ 0 & -1 \end{bmatrix}, \; G = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \; H = \begin{bmatrix} 1 & 0 \end{bmatrix} ,$$

$$J = [0], \; G_1 = [0 \; 0]^T .$$

The reference input is assumed to follow a second-order linear differential equation of the form:

$$\frac{d^2 r(t)}{dt^2} = -\omega_0^2 r(t) .$$

Substituting all these into the above design equations, the error state transition matrices become:

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -\omega_0^2 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -1 \end{bmatrix}, \; B = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

and the characteristic equation for the overall system including state variable feedback, [A−B K], can be written as:

$$s^4 + (1+K_{02})s^3 + (\omega_0^2+K_{01})s^2 + [K_2+\omega_0^2(1+K_{02})] + K_{01}\omega_0^2 = 0 \ .$$

The required gains K_1, K_2, and K_0 = [K_{01} K_{02}] can then be computed to place the four poles of the characteristic equation at designer-selected locations providing adequate asymptotic error performance and tacking of the reference input. The resulting robust feedback control system is shown in Figure 13-2[13.1]. Note the presence of an oscillator having a frequency of ω_0 radians per second as part of the feedback mechanism. This provides the controller with an internal model of the reference signal to be tracked.

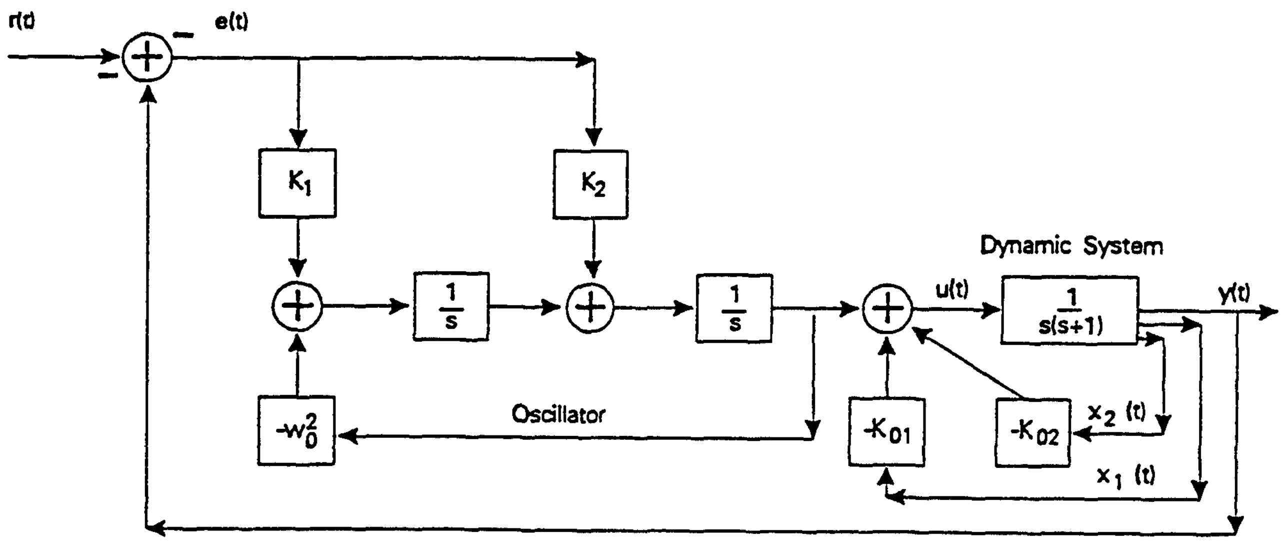


Figure 13-2. A robust feedback tracking control system.

If this same system is to track constant inputs without any steady-state error, the differential equation satisfied by the external reference input is simply:

$$\frac{dr(t)}{dt} = 0 \ .$$

The closed-loop control law can be written as before in terms of the applied control input u(t) and the system state x(t):

$$\frac{du(t)}{dt} = -K_1 e(t) - K_0 \frac{dx(t)}{dt} \ ,$$

where $K_0 = [K_{01}\ K_{02}]$. This control law can then be integrated to yield an expression for the required control input:

$$u(t) = -K_1 \int_{\tau=0}^{\tau=t} e(\tau)\, d\tau - K_0 x(t) .$$

This form is a proportional plus integral controller.

The error state z(t) will tend to zero for any and all perturbations in the underlying system parameters as long as those perturbations result in a stable system defined by the matrix $[A-BK]$. The controller design indicated above creates structurally robust blocking zeros which do not change location in the complex plane as the underlying system parameters change. These blocking zeros, located at the roots of $a_r(s) = 0$, eliminate transmission from the external signals r(t) and w(t) to the system error signal e(t).

13.4 Sensitivity Analysis

Sensitivity analysis provides a mathematical indication of a control system's viability faced with relatively small variations in the parameters of the dynamic system's mathematical model. The numerical values of these parameters are assumed to vary slightly about a set of nominal or design values. Robustness analysis, in contrast, is a mathematical technique used to investigate the viability of a control system faced with large, possibly dynamic, perturbations in the model of the dynamic system.

The mathematical models used in the design, development, and implementation of closed-loop control systems for dynamic systems involve a host of idealizations, approximations, and simplifications. A sensitivity analysis is done by the control system designer to assess the overall impact of model simplifications and modeling errors on the performance of a proposed control system design. The most common sensitivity analysis is parametric sensitivity analysis, in which the effect of a small change in the numerical value of a single model parameter, and its effect on the resulting performance of the closed-loop control system, is investigated.

The actual variation in a model parameter most often occurs in practice as a result of manufacturing or assembly tolerances or unanticipated ageing, wear and tear, or abuse. The effect of this parameter variation is to alter the transfer function of the closed-loop control system and thereby adversely affect the stability or performance of the system. A prudent designer will strive to anticipate parameter sensitivities and include them in the design of the feedback system. If the parameter is subject to the direct control of the designer, the selection of an appropriate nominal value can be included in the system design procedure.

Bode defined the sensitivity of a transfer function, G, to one of its parameters, k, as the ratio of the percent change in k to the percent change in G. This can be mathematically written as:

$$S_k^G = \frac{\left[\dfrac{\partial k}{k}\right]}{\left[\dfrac{\partial G}{G}\right]} .$$

If the parameter k changes by one percent, the sensitivity S_k^G indicates the expected percent change in G.

It is today more common to use the inverse of this expression for S, and also to consider several additional sensitivity measures[13.11].

The sensitivity of G with respect to the parameter k is defined as:

$$S_k^G = \frac{\left[\dfrac{\partial G}{G}\right]}{\left[\dfrac{\partial k}{k}\right]} = \frac{\partial \ln(G)}{\partial \ln(k)} = \frac{k}{G}\left[\frac{\partial G}{\partial k}\right] .$$

Similarly, the sensitivity of the phase angle of G with respect to the parameter k is defined as:

$$S_k^{\phi_G} = \left[\frac{\partial \phi_G}{\partial k}\right] ,$$

and the sensitivity of the magnitude of G with respect to the parameter k is defined as:

$$S_k^{|G|} = \left[\frac{\partial\,|G|}{\partial k}\right] ,$$

When two or more alternate structures are available to implement the transfer function G, a careful sensitivity analysis will indicate that structure having the lowest sensitivity and the one that is most suitable in terms of this measure.

A simple example can be constructed to illustrate the value and application of sensitivity analysis. Figure 13-3[13.1] shows an open-loop system and that same system with a closed-loop controller.

For the open-loop system with forward transfer function G_1, the transform of the output, C, equals the product of the transform of the input, R, and the forward transfer function:

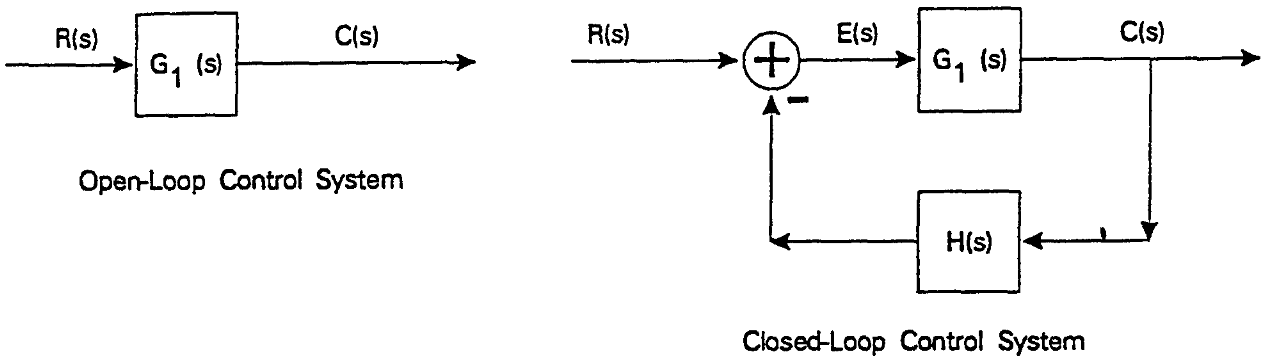


Figure 13-3. Open- and closed-loop control systems.

$$C = G_1 R .$$

A small change δG_1 produces a corresponding small variation in the output:

$$\delta C = R \delta G_1 .$$

The transfer function in this case is:

$$T = \frac{C}{R} = G_1 .$$

and the sensitivity of T to a small change in G_1 is:

$$S_{G1}{}^{T} = \left[\frac{G_1}{T}\right] \left[\frac{\partial T}{\partial G_1}\right] = (1)(1) = 1.$$

Thus any small change in the forward transfer function G_1 immediately and entirely affects the output of the system.

For the closed-loop system with forward transfer function G_2 and feedback transfer function H, the overall transfer function T is:

$$T = \frac{C}{R} = \frac{G_2}{(1+G_2 H)} .$$

and the sensitivity of T to a small change in G_2 is:

$$S_{G2}^{T} = \left[\frac{G_2}{T}\right] \left[\frac{\partial T}{\partial G_2}\right] = \left[\frac{1}{1+G_2 H}\right].$$

As the loop gain G_2H increases, the sensitivity of T to a small change in G_2 decreases. This demonstrates the reduced sensitivity of a feedback control system to parameter variations in the forward transfer function or underlying controlled dynamic system.

13.5 Summary

A closed-loop control system must be regulated even if its mathematical model is slightly altered. Regulation is achieved in a system with good disturbance rejection when the system output remains close to the input even when the input is zero. A low sensitivity system maintains good regulation in spite of changes in the dynamic systems parameters. A robust control system combines both good disturbance rejection and low sensitivity. Design approaches to regulation, robustness, and sensitivity are presented.

REFERENCES

13.1 Franklin, G. F., Powell, J. D., and Emami-Naeini, A., Feedback Control of Dynamic Systems, Addison-Wesley Publishing Company, Reading, Massachusetts, 1986.

13.2 Rosenbrock, H.H., Computer-Aided Control System Design, Academic Press, London, 1974.

13.3 Zames, G., On the Input-Output Stability of Time-Varying Nonlinear Feedback Systems - Part I: Conditions Using Concepts of Loop Gain, Conicity, and Positivity, IEEE Transactions on Automatic Control, Vol. 11, pp. 228-238, 1966.

13.4 Safonov, M.G., Robustness and Stability Aspects of Stochastic Multivariable Feedback System Design, Ph.D., Thesis, Massachusetts Institute of Technology, 1977.

13.5 Kwakernaak, H., Optimal Low-Sensitivity Linear Feedback Systems, Automatica, Vol. 5, pp. 279-286, 1969.

13.6 Doyle, J.C., Guaranteed Margins for LQG Regulators, IEEE Transactions on Automatic Control, Vol. 23, pp. 756-757, 1978.

13.7 Doyle, J.C. and Stein, G., Robustness with Observers, IEEE Transactions on Automatic Control, Vol. 24, pp. 607-611, 1979.

13.8 IEEE Transactions on Automatic Control, Special Issue on Linear Multivariable Control Systems, Vol. 26, 1981.

13.9 Davison, E.J., Design of Modern Control Systems, In: Bill, D.J., Cook, P.A., and Monroe, N., eds., Peter Perigrinus, Stevenage, UK, pp. 246-273, 1982.

13.10 Davison, E.J., and Gesing, W., The Systemstic Design of Control Systems for the Linear Multivariable Servo-Mechanism Problem, In: Sain, M.K., Peczkowski, J.L., and Melsa, J.L., eds., Alternatives for Linear Multivariable Control, National Engineering Consortium, Chicago, pp. 257-300, 1978.

13.11 Brogan, W. L., Modern Control Theory, Second Edition, Prentice Hall, Inc., Englewood Cliffs, New Jersey.

CHAPTER 14
PRECISION GUIDED MUNITIONS

14.1 Overview Chapters

The previous chapters of this report have outlined selected mathematical tools and techniques of modern control theory applicable to the analysis, design, and implementation of guidance and control systems for tactical weapons. Up to this point, these tools have been discussed in general terms, and highly simplified examples have been presented to acquaint the reader with these methods and their application.

The objective driving the application of modern control theory to the guidance and control of tactical guided weapons is increased effectiveness. An efficient trajectory is sought, starting from a launch point and ending with an impact with the target. Once that trajectory is determined, a means for steering the weapon along the trajectory must be developed and implemented. In the following sections, an overview is provided about how various munition systems utilize guidance and control.

The nomenclature associated with tactical guided weapons is varied, overlapping, and confusing. The Army, Navy, Marines, and Air Force each have their own terminology. Foreign weapons are also referred to differently, especially by the intelligence community. This proliferation of nomenclature often gives no clue as to how the weapon is guided or controlled.

Precision guided munition (PGM) is used as a generic term to include all tactical guided weapons. Before describing the many alternative ways of naming PGMs, a brief description will be given of a generic PGM. A PGM is a munition[14.1] that can change its direction, or react, in order to hit its target, based upon information obtained during its flight. This information is obtained and processed by the PGM's guidance system and the reaction of the PGM to this information is performed by the PGM's control system. The hardware and software components used to determine an efficient trajectory and to steer the PGM toward its target comprise the integrated guidance and control system. Many specialized guidance and control systems are in use, each tailored to the particular characteristics of the associated PGM. The complexity of any one system is determined by factors such as the type, speed, and location of the target, the maneuverability of the PGM, the precision required for weapon delivery, and the environmental conditions faced by the system.

The primary functions of the guidance and control system are sensing, information processing, and control action. Guided missiles, guided projectiles, or other similar weapons must be steered from a short time after launch until the time of target impact or interception. The position, motion, and type of target must be sensed by some means, and this information, together with the present location of the tactical weapon and a knowledge of its maneuverability, must be rapidly processed to define an efficient trajectory.

The guidance and control system generates steering commands which cause deflections of aerodynamic control surfaces or changes in one or more thrust vectors. These deflections in turn cause the missile or weapon to maneuver along a trajectory which eventually brings the weapon closer to its target. The mechanism by which a steering command is converted into a maneuver is normally implemented in the guidance and control system by means of a predetermined guidance law or control algorithm.

Figure 14-1 illustrates the major components of a missile guidance and control system. The inner control loop in this figure is the guidance loop. The guidance loop contains one or more sensors used to determine the relative motion of the missile and the target. This may be done by determining only the motion of the missile and comparing the missile location to that of a fixed target, or by determining the motion of both objects.

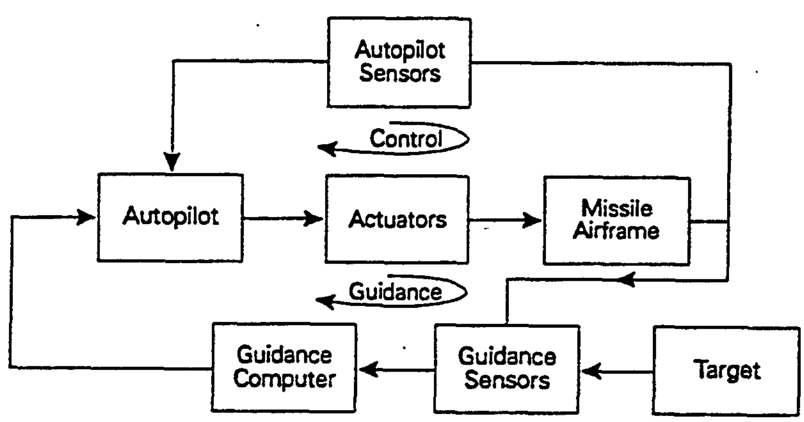


Figure 14-1. Functional block diagram for a generic guided missile system.

The motion sensors play the role of an error detector in a conventional feedback control system. The error between the missile's position and that of the target is computed and used to produce an error signal.

The motion information is processed in the guidance computer, which may be either an analog device or a digital computer executing a predetermined guidance program. The guidance computer generates lateral acceleration commands for the autopilot. A lateral acceleration is an acceleration at right angles to the present missile heading. The autopilot's function is to control the pitch, yaw, and roll orientation of the missile. The autopilot processes the command accelerations generated by the guidance computer and stabilizes the missile during its flight. The autopilot is a feedback control system and its complexity depends on the missile configuration and aerodynamics.

14.2 Classification of PGMs by Launcher-Target Locations

The simplest scheme to classify different PGMs is to identify where the launch platform is located and where the target is located. This scheme is most often applied to missiles, as follows:

- surface-to-air missiles (SAMs) launched from the surface of the earth against an airborne target
- surface-to-surface missiles (SSMs) launched from one point on the earth's surface for use against a target at a second point on the surface
- air-to-surface missiles (ASMs) launched from an aircraft or other aerial platform against a target on the earth's surface
- air-to-air missiles (AAMs) launched from one aircraft or other aerial platform for use against a second similar aerial target

Unfortunately, variations of this approach are possible. The Air Force prefers to call ASMs, AGMs, air-to-ground missiles.

14.3 Classifications of PGMs by Target Type

Very often the target being attacked controls the primary characteristics of the PGM regardless of the launch platform or its location. Consequently, PGMs may be classified as antitank or antiship. Other categories include:

AAW	Anti-Air-Warfare whether by AAM or SAM
USW	Undersea Warfare by surface ships, aircraft, or submarines
HTM	Hard Target Munitions for defeating hardened or buried targets
ARM	Anti-Radiation Missile that homes on emitting enemy air defense radars

14.4 Classification of PGMs by Sensor Operation

Modern PGMs use a variety of sensors on the launch platform as well as carried by the PGM itself to perform their mission. Descriptions of different sensor operations are given for terminal homing of missiles but may apply to other munitions as well.

Active Homing. A missile with an active homing system carries its own transmitter, a source of illuminating radiation, and a receiver tuned to that radiation (see Figure 14-2). Active homing systems are normally implemented by a radar system operating in the microwave or millimeter wave region of the electromagnetic spectrum. The maximum transmitter power and antenna dimensions are constrained by the available volume and area allocated to the radar sensor. This limits the target acquisition range of the seeker. If the seeker has been designed to autonomously search for, detect, and lock on to a target after being launched in the general direction of an expected target, the stand-off range of the missile can be considerably increased. Some missiles employ a mid-course guidance phase during which a predetermined trajectory is followed or the radar is operated in a semi-active mode. In a semi-active mode the illuminating source is located at a point other than onboard the missile.

Active homing radar seekers have been applied to air-to-air, air-to-surface, and surface-to-surface missiles. When operated with a coherent pulse waveform, a radar seeker can provide range and range rate homing data in addition to the angular line-of-sight rate. Radar seekers have been applied to target applications requiring an all-weather fire-and-forget capability.

Millimeter wave radar seekers have received increased attention in recent years. This stems from the high-range resolution, angle resolution, and range rate resolution theoretically achievable in the millimeter wave spectral region. The application of millimeter wave seekers is expected to increase the radar homing capabilities of tactical missiles for use against land-based surface targets such as tanks. Synthetic aperture techniques are also being applied to permit the application of high-resolution ground-mapping methods as a means for mid-course guidance and the attack of high-value fixed installations.

Semi-Active Homing. In a semi-active homing system the target is illuminated by a radar or a laser target designator located on the ground or onboard the launch aircraft. The missile seeker contains a sensor tuned to the illuminating signal. This sensor may be a radar receiver or an electro-optic detector. The sensor collects the energy reflected from the target and processes the resultant signal to determine the relative angular position of the target. This angular information is then used as an input to the guidance computer.

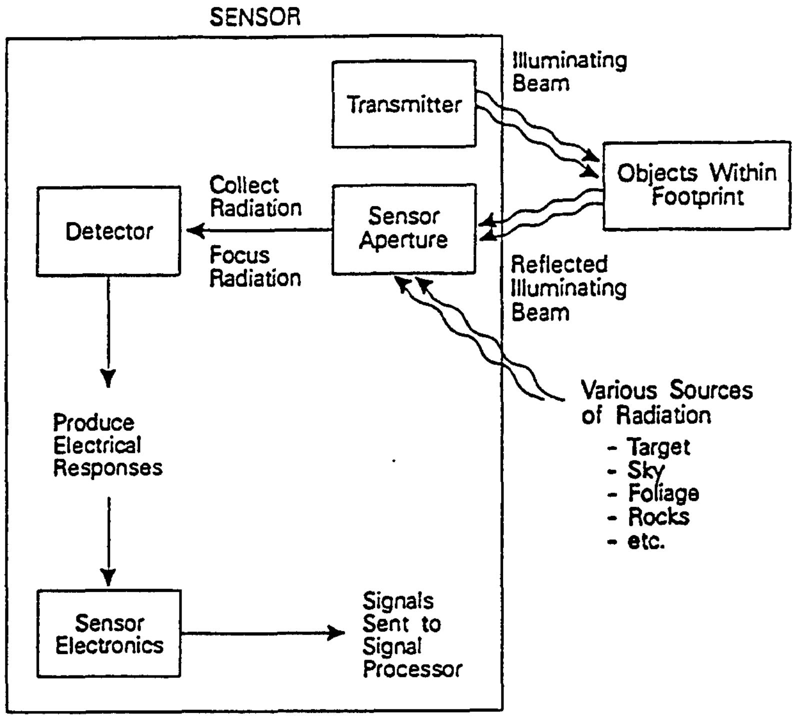


Figure 14-2. Operational block diagram of an active sensor.

The atmospheric attenuation as a function of wavelength is shown in Figure 14-3. There are several windows of relatively low attenuation which are suitable spectral regions in which to operate any active or semi-active homing device. Although a laser operating in the near IR region at a wavelength of 1.06 μm is restricted to operation in relatively good weather and short ranges due to atmospheric attenuation, this laser's pulse repetition capability makes it suitable for use against surface targets. A launch-and-leave capability can be obtained by the use of this technique in an ASM system. The target can be designated by an operator onboard the launch aircraft, and that aircraft can remain at a relatively long stand-off range during the engagement.

Semi-active radar homing methods offer an all-weather capability, but this technique is normally restricted to use against aerial and naval targets due to the effects of ground clutter. Work is underway to improve the performance of millimeter wave radar systems for use against ground targets by the use of more sophisticated signal processing techniques. When used against low-altitude aerial targets the target can be distinguished from ground clutter by means of the Doppler shift in the received radar signal. A coherent radar system requires that a reference signal be provided to the missile radar receiver. This is accomplished by the addition of a rear-looking antenna and the use of either a continuous or an interrupted radar waveform. A pulse waveform can be effectively applied against crossing aerial and ship targets.

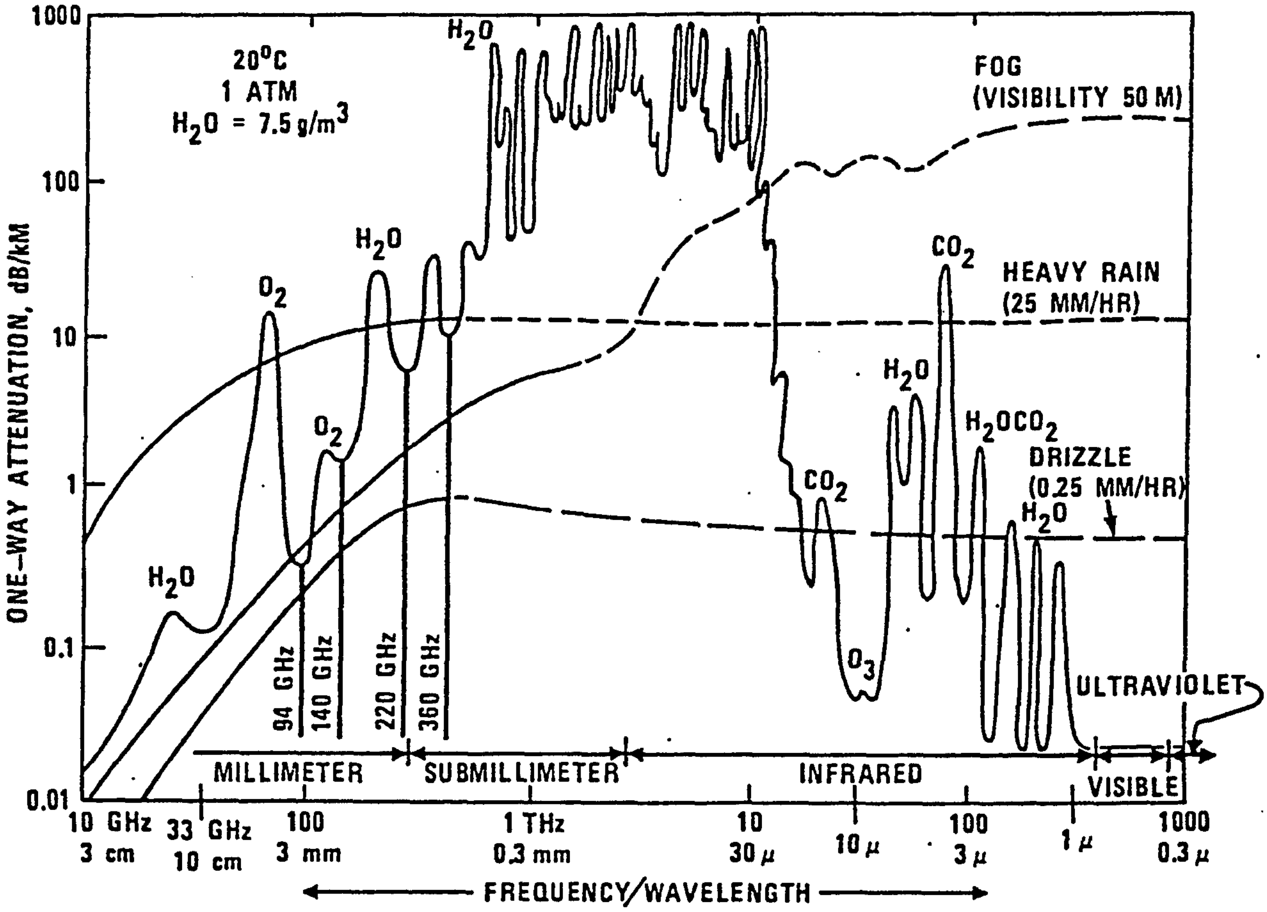


Figure 14-3. Attenuation by atmospheric gases, rain, and fog.

<u>Passive Homing Missile Systems</u>. In a passive homing missile system the primary energy source whose energy is received by the missile seeker is either the effect of the sun on the target or self-generated target emissions. There may be other sources as indicated in Figure 14-4. The energy source in a passive homing missile system is not part of the guidance system, and is not controllable by the weapon system operator. Passive homing systems are designed to have a fire-and-forget capability. Systems operating in nearly all possible spectral regions have been proposed, including missile seekers which combine energy received in several spectral regions as a means for improving target detection and discrimination capability.

The specialized anti-radiation missile (ARM) is designed for use against enemy communication systems and enemy radar transmitters. These missile seekers contain one or more antennas which provide directional information. A radar guided missile can also be designed to operate in a similar home-on-jam method. In that case the missile seeker tracks the source of the jamming energy rather than the source of reflected radar signals.

Millimeter wave devices have been proposed with dimensions suitable for inclusion in small missiles, submunitions, and cannon-launched projectiles. By including a complete millimeter wave radar system, operation in an active mode for autonomous target detection is possible, and the system can switch to a passive radiometric mode for the terminal phase of the engagement. In this way the potential adverse affects of radar glint may be avoided and the expected miss distance may be reduced.

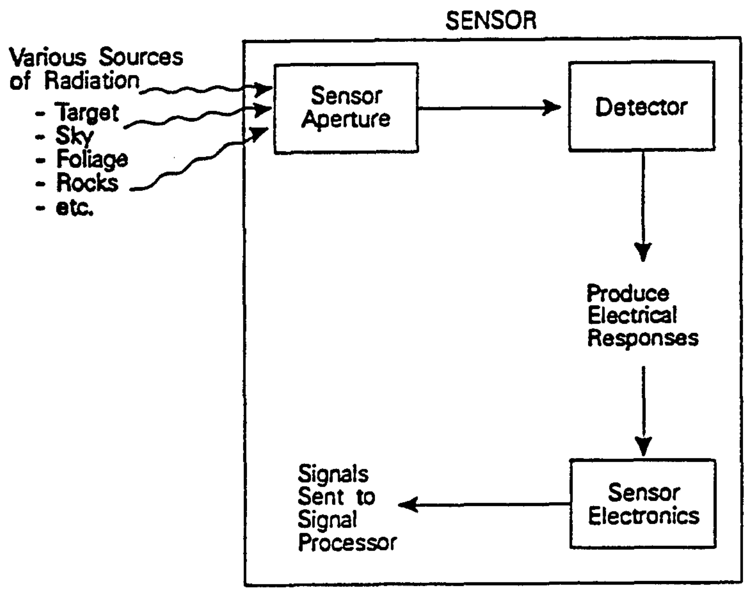


Figure 14-4. Operational block diagram of a passive sensor.

Electro-optical guided weapons may be equipped with a seeker which operates in an imaging mode. In the visual region, standard and low-light-level television sensors can be applied. In the 3-5 μm middle IR spectrum or in the 8-12 μm far IR spectrum, thermal imaging sensors can be utilized. These spectral regions correspond to windows of low atmospheric attenuation. An imaging seeker responds to the difference in contrast between the target and its background. A thermal imaging seeker responds to the temperature-dependent IR radiation of the target and the background.

IR imaging seekers have a day-night capability since they depend on the target and background emissions rather than on a direct source of illumination as required by a television-based seeker. Imaging IR seekers are known to have somewhat better performance under poor battle field conditions, smoke, dust, and adverse weather compared to television-based systems. The sensitivity and angular resolution of an imaging IR seeker is somewhat better than that of a millimeter wave radiometric seeker when both operate in good weather. Scanned focal point arrays of IR sensors can offer especially high angular resolution.

The target detection and tracking capability of all imaging type seekers, both visual and IR, depends on the nature of the target signature compared to the background of the scene and the algorithms used for detection and tracking. In most applications, methods drawn from image processing and pattern recognition must be implemented to provide effective seeker operation. These methods may include thresholding, edge detection, centroid computation, and area correlation.

IR seekers have classically been designed as hot-spot trackers. In these systems, the seeker is designed to track an IR radiating source using only a few detector elements. Target detection and tracking requires some means of optical modulation, either a reticle placed in the focal plane, or a mechanical scanning process in which the total field-of-view is scanned by a single detector element having a small instantaneous field-of-view. Target discrimination and countermeasure immunity can be improved by adding a multispectral capability. Hot-spot trackers have been applied to short-range SAM and AAM systems in which the IR source is the aircraft exhaust pipe or plume. For high-speed aerial targets the aerodynamically heated leading edges may also serve as hot-spot targets.

14.5 Classification of PGMs by Munition Type

Improvements in guidance and control technology can be applied to almost every kind of weapon, explosive device, or munition. Missiles, projectiles, bombs, rockets, land mines, sea mines, and torpedoes can all have sensors, position orientation capabilities, propulsion, and guidance-aided fuzing. Wide area mines, sensor-fuzed weapons, and guided bombs are all variations on the theme of smart munitions.

14.6 Classification of PGMs by Formal Military Designations

According to Department of Defense Directive (DoDD) 4120.15, the Military Services must formally designate PGMs[14.2] with the following terminology. The prefixes may not be acronyms.

AIM	Aerial Intercept Missile such as the air-to-air AIM-9 Sidewinder or the AIM-120 Advanced Medium-Range Air-to-Air Missile (AMRAAM)
AGM	Air-to-Ground Missile such as the AGM-65 Maverick or AGM-88 High-Speed Anti-Radiation Missile (HARM)

MIM	Mobile Air Intercept Missile such as the surface-to-air MIM-23 Homing All the Way Killer (HAWK) or MIM-104 Patriot
MGM	Mobile Surface Attack Missile such as the surface-to-surface MGM-31 Pershing or MGM-52 Lance
LGM	Long-Range Guided Missile such as the LGM-25 Titan or the LGM-30 Minuteman
GBU	Guided Bomb Unit such as the GBU-10 EO Guided Bomb GBU-15(V) 1/B and 11R Guided Bomb GBU-15(V) 2/B
RIM	Ship launched air Intercept Missile such as the surface-to-air RIM116A Rolling Air Frame Missile (RAM)
BGM	Ground launched Guided Missile such as the BGM-71A Tube Launched Optically Tracked Wire Guided (TOW)
M-712	Copperhead tube launched indirect fire laser guided projectile

14.7 Classification of PGMs by Capabilities

The Army has developed a classification scheme[14.3] for PGMs that is a mix of other approaches plus incorporation of the maturity of the technology possessed by the guidance and control components. Precision Guided Munitions consist of three subsets as indicated by Figure 14-5.

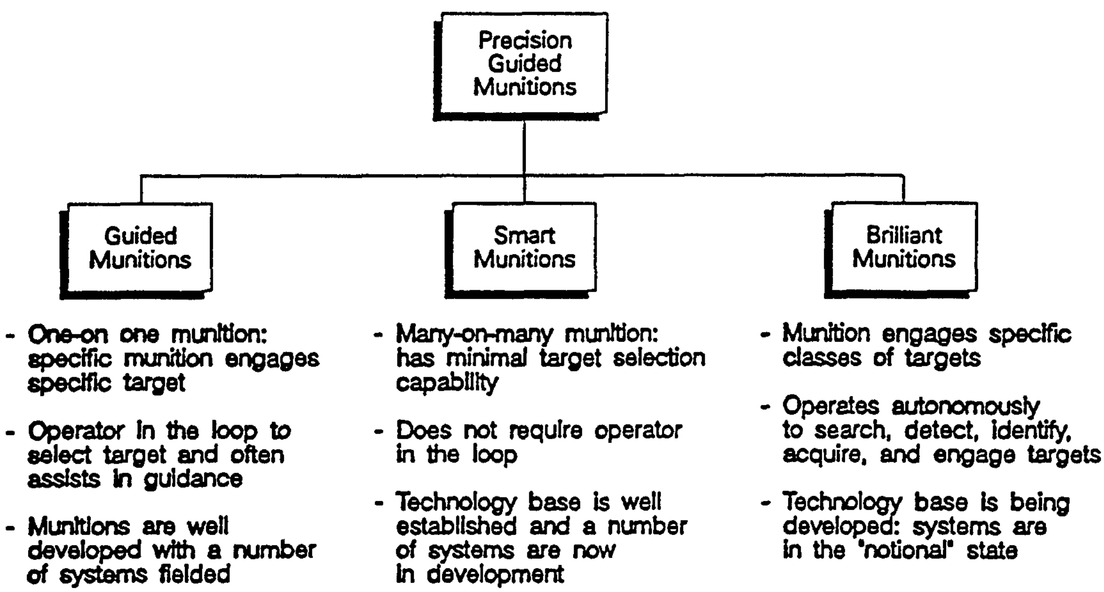


Figure 14-5. Classes of PGMs.

Guided munitions are characterized as one-on-one munitions that require an operator in the loop to function. Each munition is directed to a specific target by the operator or a gunner. This requires a direct line-of-sight (LOS) between the operator (or the sensor being used by the operator) and the target. Systems of this class are already fielded. SMs are in the development phase with weapon fielding scheduled for the 1990s. Brilliant munitions are in the notional state. It is conceived that such munitions would operate autonomously, as do SMs, but they should be capable of selectively identifying and engaging specific target sets. It will be a number of years before brilliant munitions are ready for development.

Smart munition weapons are viewed as an addition to, not a replacement for, guided munitions weapons. Guided munition weapons have the distinct advantage of precisely engaging specific targets. In close battle, where friendly and enemy forces are intermingled, the ability to engage specific targets is essential. Conversely, wherever they can be delivered, smart munition weapons are most effective against high-target densities. Guided munition weapons and smart munition weapons are complementary.

Smart munitions are further defined in a sub-subset of PGMs as indicated in Figure 14-6 Terminally guided munitions (TGMs) are hit-to-kill weapons; they guide to the target and an on-board warhead is fuzed upon target impact. Sensor fuzed munitions (SFMs) are shoot-to-kill weapons; the warhead is fuzed some distance (tens of meters) from the target while the munition is aimed at the target.

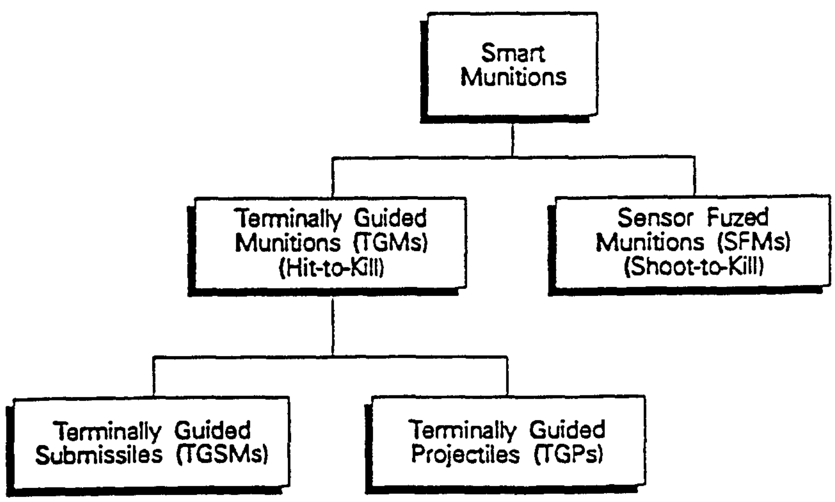


Figure 14-6. Types of smart munitions.

In the past, there was little distinction between the two types of TGMs, terminally guided submissiles (TGSMs) and terminally guided projectiles (TGPs). TGSMs were delivered by missiles or rockets while TGPs were delivered by cannons. The difference being simply that TGPs had to survive high cannon launch accelerations of thousands of g's, while TGSMs faced low-launch accelerations of, at most, tens of g's. However, the Army is now pursuing the development of TGPs with conventional geometry, i.e., a TGP having similar size and weight as that of a conventional artillery round. While this feature will greatly enhance tactical utility, TGPs are becoming considerably different from TGSMs. Size and weight considerations normally preclude delivery of TGPs from missiles or rockets. They are only delivered one at a time from cannons, and the size-weight constraint is that dictated by ballistic requirements. However, several TGSMs may be delivered by a single missile or rocket. Though the requirements and designs of TGPs and TGSMs are rapidly diverging, they share a well-founded and common technology base.

14.8 Classification of PGMs by Range

Nature has created a set of range bins for PGMs. Short range includes line-of-sight, the limit of human vision to detect targets, or about 3 to 5 miles. Medium range goes to the horizon or about 10 to 15 miles. Long range is anything beyond. One marker for long range for the Army is the range of mobile artillery of about 20 to 30 miles. Longer ranges become the domain of tactical and strategic missiles. It is these range bins and the types of target being attacked that most affect guidance and control systems.

14.9 Classification of PGMs by Direct Guidance

When the target is moving it is necessary to implement a direct guidance method for the terminal phase of the engagement. A direct guidance method uses updates of the target position to revise the missile's trajectory. A sensor contained in the missile seeker maintains line-of-sight contact with the target and provides an indication of the target's relative angular position, range, and velocity. This sensor may be initialized by a weapon operator prior to missile launch, or the missile may be autonomous, having the ability to search for and detect a target without operator intervention.

A wide array of signal processing techniques have been implemented in attempts to discriminate the target from its background, other neighboring targets, and potential decoys. Various spectral filters are used to extract the geometric and kinematic variables required for implementation of target tracking and missile guidance. Low-pass filters are commonly used to attenuate high-frequency noise contained in the sensor signal. The design of these filters is based on an assumed knowledge of the target and noise signals.

Applications for modern control theory include state variable modeling and analysis for the missile and target aerodynamics and kinematics, the estimation of target motion state variables based on observed sensor data, the development of optimal trajectories and maneuvers, adaptive control of the missile in the event of hardware failures or damage in flight, and regulation of the attitude of both the missile and the associated tracking platform. For example, optimal state estimators such as the Kalman filter can be used to separate the target from the noise or background by developing updated information about the target and missile dynamics and the noise covariance matrices.

The use of a direct guidance method which accounts for the relative motion between the target and the missile makes it possible for the missile to hit a moving target. The way in which the error signals representing the relative positions of the missile and the target are generated is determined by a guidance law. The guidance law is a mathematical representation of the relationship between the sensed target and missile information. The guidance law is implemented in the guidance computer. The guidance computer may be an electronic analog device or a digital computer equipped with analog and digital inputs and outputs.

Direct guidance methods can be further classified into one of three categories: command, homing, or beamrider guidance. Each category implements a unique closed-loop system containing the missile, the target, and possibly an operator. Each category of direct guidance provides ample opportunities for the application of both classical and modern control theory and technology.

Command Guidance. In a command guidance system the missile guidance commands are generated in a guidance computer located external to the missile. The missile and the target are both tracked by one or more sensors located on the earth's surface or on an aerial platform or aircraft. The sensed information is used to determine a guidance command which is then transmitted to the missile by means of an RF link, a wire, or an optical fiber.

A variation of this approach uses a sensor carried on-board the weapon. This sensor transmits line-of-sight information about the target back to the guidance computer. This technique eliminates the need to directly track the target by means of an external device, and can provide a measure of immunity against countermeasures.

Figure 14-7 depicts generic command guidance based upon separate target and missile tracking radars, with the command link included in the missile tracking radar. The solid arcs emanating from the tracking radars represent the transmitted radar signals. These signals are reflected by the target in all directions, the dashed arcs representing the reflection in the direction of the target tracking radar. This reflected signal is processed by the target tracking radar to determine the target's range and

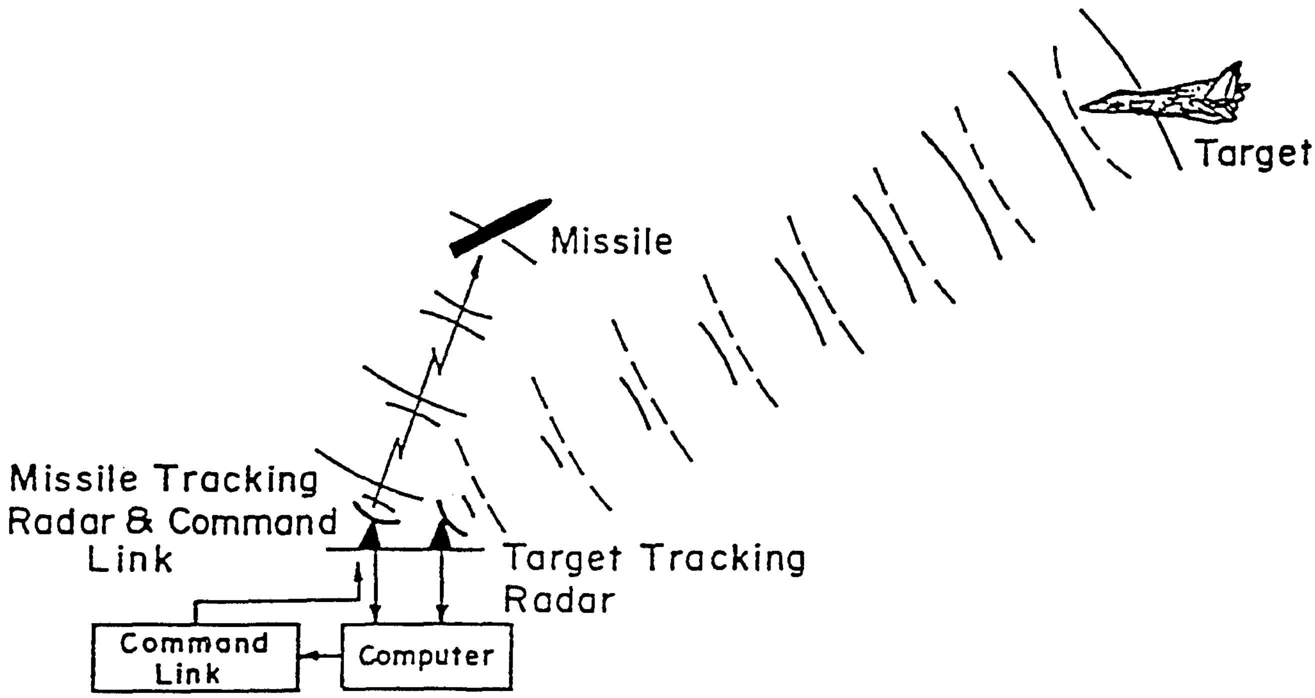


Figure 14-7. Generic command guidance (surface-to-air case).

angle and the information is fed into the guidance computer. The missile carries a beacon, or radar transmitter, which is triggered by the signal received from the missile tracking radar. The beacon transmits a strong signal back to the missile tracking radar, shown by the solid arcs emanating from the missile. This insures accurate tracking of the missile and can also provide a data link from the missile to the ground. The missile's range and angle is also fed into the computer. The computer then calculates the trajectory the missile should fly in order to intercept the target and generates commands which are sent to the missile via the command link. This link is shown by the jagged line between the missile tracking radar and the missile. By monitoring the target and missile positions and refining the trajectory calculations throughout the engagement, the missile is guided to intercept the target.

The guidance computer has complete knowledge of the engagement geometry and is programmed to select and implement the optimal trajectory along which to guide the missile. In the terminal phase of this engagement, either a collision or a constant bearing course might be implemented. In either case the missile is steered to a predicted impact point. The actual impact point depends on the true target and missile positions, velocities, and orientations.

Command-to-Line-of-Sight (CLOS) Guidance. CLOS guidance is an implementation of the three-point LOS guidance law. In a CLOS guidance system the missile is steered to the target along the LOS joining the guidance command point to the target. The guidance command point in a CLOS system is usually on the ground. The target is tracked from the guidance command point by a radar or an electro-optical system. The missile is tracked by a similar collocated system or by a separate missile tracker.

The target and missile sensors provide angular input data to the guidance computer. The guidance computer processes this angular data to obtain the azimuth and elevation displacement of the missile from the LOS joining the guidance command point and the target. The angular errors in radians are multiplied by an estimate of the instantaneous missile range. This computation results in an estimate of the linear distance that the missile is off course. The lateral acceleration commands required to reduce this lateral distance are computed in a way which guarantees stability of the closed-loop guidance process and provides a fast compensating response for linear displacement errors.

An anticipatory control system is required, since the linear displacements are corrected by means of a lateral acceleration command. To stabilize this guidance loop a filter is required which separates the LOS measurement data from the system noise.

The accuracy of the CLOS guidance loop, measured by the steady-state error in response to a specified command input, can be improved by adding feed-forward compensation commands to the error compensation commands. The feed-forward command implements the lateral acceleration required to just keep the missile on the rotating LOS course from the command point to the target. This acceleration command is derived based on measurements of the LOS rotation rate, estimates of missile range, and estimates of missile velocity and acceleration.

A functional block diagram of a semiautomatic CLOS missile system is shown in Figure 14-8. The illustration shows the feed-forward guidance loop and the feedback guidance loop. The infrared (IR) sensor is bore-sighted with the optical axis of the target tracker, and this sensor determines the angular deviation of the missile from the LOS.

When additional information is available to the guidance computer other guidance laws beyond the three-point guidance law can be implemented. This makes it possible to optimize or shape the missile trajectory so as to reduce the required lateral accelerations or to project a trajectory which allows the missile to approach the target from the most favorable direction.

CLOS guidance systems have been implemented for short-range antitank and air defense weapon systems. The guidance accuracy of an operator-assisted CLOS system decreases somewhat

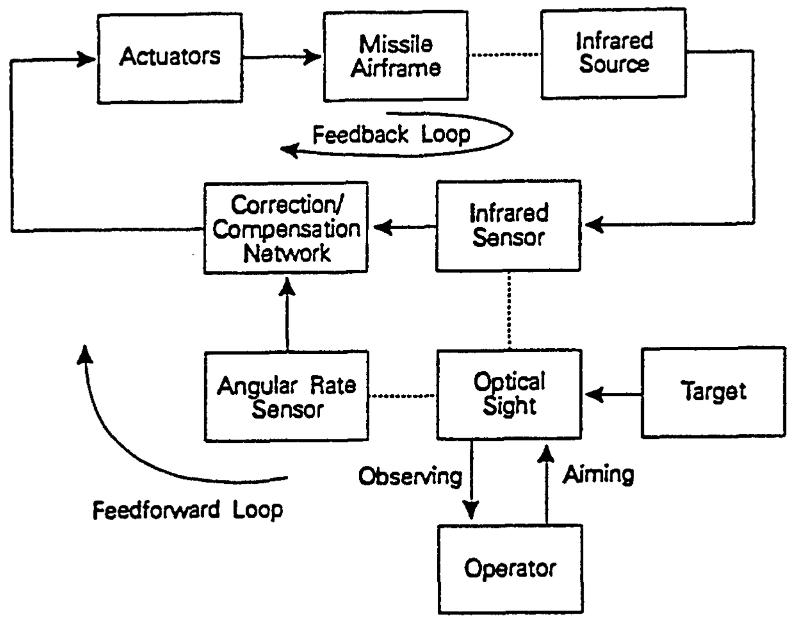


Figure 14-8. Semiautomatic CLOS for an antitank missile.

with increasing range to the target. The presence of an operator in the control loop also limits a CLOS system to the engagement of a single target.

Homing Guidance Systems. A homing guidance system uses a sensor mounted in the seeker portion of the missile to provide an on-board guidance and control system with information concerning the relative missile-target motion. The sensor tracks the target by means of reflected or radiated energy in the microwave, millimeter wave, IR, visible, or ultraviolet portions of the electromagnetic spectrum. Two or more of these regions may be used simultaneously, and combinations of active and passive multispectral sensor systems have also been investigated. The performance of a homing missile system depends on the location of the primary source of energy in a passive homing system or the location of the transmitter in an active system. This energy source may be located in the missile, on the target, or external to both the missile and the target.

The geometry of a homing seeker subsystem is illustrated in Figure 14-9. If the sensor is fixed to the missile body (a strapdown or body-fixed seeker) the sensor can determine only the relative LOS to the target. A gimballed inertially stabilized seeker can also provide the LOS rate. Radar seekers are used to provide range and range rate in addition to angular data. In addition, the implementation of any one guidance law almost always requires some additional information, for example, the missile body airframe motion in terms of pitch and yaw rates. This additional data must be provided by appropriate rate sensors.

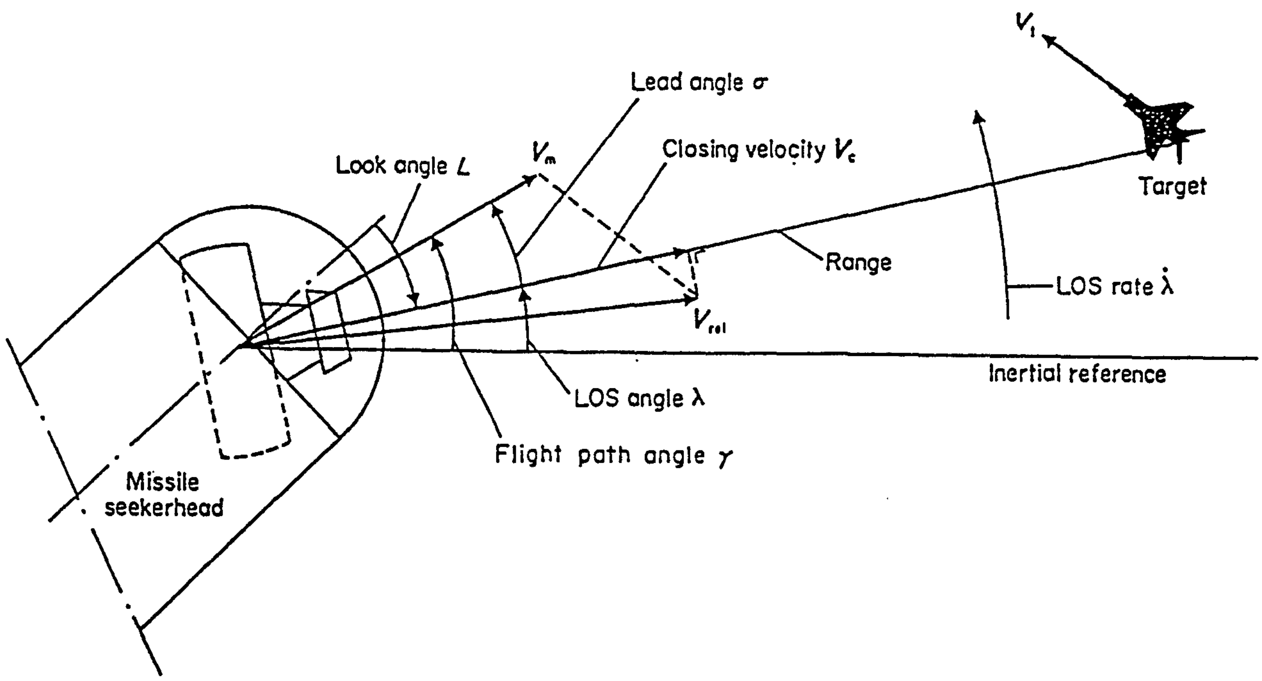


Figure 14-9. Homing guidance.

14.10 Classification of PGMs by Guidance Laws

Most present guided weapons use classical guidance laws based on controlling the lateral acceleration of the weapon perpendicular to its longitudinal axis. The function of the missile's control system is to execute the lateral acceleration commands generated by the guidance computer.

There are two main approaches toward implementing a homing or two-point guidance law in which only relative information about the missile and the target is used. The first approach is pursuit guidance, and the second is proportional navigation. In a pursuit guidance system the sensor measures the angle between the missile body vector (for body pursuit) or the missile velocity vector (for velocity pursuit) and the LOS to the target. In pursuit guidance this angle is driven to zero or some small constant value (for modified pursuit guidance). Modified pursuit guidance, in which a small preset lead angle is used, allows the missile to anticipate the speed of the target.

In a proportional navigation guidance system the LOS angular rate, measured with respect to an inertial reference, is driven to zero. This is accomplished by setting the flight path turning rate proportional to the sensed LOS rate. As a result, the relative missile-to-target velocity is aligned with the LOS from the missile to the target. Proportional navigation or a variation of this basic method has been used in nearly all guided weapons. Pursuit guidance is easier to implement in analog hardware than proportional navigation, and is somewhat less sensitive to system noise than proportional navigation, but experience has shown pursuit guidance to be effective only when restricted to stationary or slow moving targets. Table 14-1[14.4] summarizes the parameters of the most common homing guidance laws.

TABLE 14-1. BASIC HOMING GUIDANCE LAWS

GUIDANCE LAW	LATERAL ACCELERATION DEMAND, $V_m \frac{d\lambda}{dt}$
Pure pursuit, $\sigma = 0$	$K\sigma$
Modified pursuit, $\sigma = \sigma_p$	$K(\sigma - \sigma_p)$
Body pursuit, L = 0	KL
Proportional navigation, $\frac{d\lambda}{dt} = 0$	$V_m K_n \frac{d\lambda}{dt}$
Corrected proportional navigation, $\frac{d\lambda}{dt} = 0$	$K_e V_c \frac{\frac{d\lambda}{dt}}{\cos(L)}$
Extended proportional navigation, $\frac{d\lambda}{dt} = 0$	$K_e V_c \frac{\frac{d\lambda}{dt}}{\cos(L)} - V_m \tan(L)$

where

$\frac{d\sigma}{dt}$ is the LOS angular rate,

V_c is the closing velocity,

V_m is the missile velocity,

K, K_n, and K_e are gains,

$\frac{dV_m}{dt}$ is the missile longitudinal acceleration,

σ_p is a preset squint angle,

$\frac{d\lambda}{dt}$ is the missile flight-path turning rate,

L is the look angle of the seeker axis to the target.

Guidance laws are implemented in the way the PGM airframe is controlled.

A mono-wing PGM is aerodynamically designed so that it has an optimal lift-to-drag ratio in the plane perpendicular to the wing surface. A mono-wing PGM is controlled by the bank-to-turn method. The PGM increases its elevation angle to increase the lift of the wing, and then performs a roll maneuver to orient the lift vector in the direction demanded by the guidance computer. The bank-to-turn method is preferred for mono-wing configuration long-range cruise missiles and for missiles that require large lateral accelerations in their efforts to follow target maneuvers.

A cruciform missile, having a symmetric structure and two identical pitch and yaw planes of control, is controlled by the skid-to-turn method. The commanded lateral acceleration is decomposed into separate pitch and yaw commands. These commands are executed simultaneously by the pitch and yaw control channels. The skid-to-turn method is used in most missile designs and is attractive because it employs two identical control channels and serves a symmetrical missile design.

The autopilot is responsible for the fast and accurate execution of acceleration commands over a wide range of flight conditions. The autopilot provides a means for stable, responsive closed-loop control of the missile. The autopilot forms a link between the guidance computer and the missile airframe. The feedback signals available to the autopilot include sensed airframe lateral acceleration and angular rate. The autopilot is designed to improve the speed of response of the airframe, reduce transient overshoot, and maintain a nearly constant autopilot/airframe gain.

Roll attitude stabilization or a roll attitude command channel may be incorporated into the design of an autopilot. Open-loop control is occasionally used to reduce cost and complexity in the design of small tactical missiles. Adaptive methods or gain scheduling are sometimes used to compensate for variations in the dynamic pressure. The dynamic pressure is given by $1/2\ \rho\ V_m^2$, where ρ is the atmospheric density at the altitude of operation and V_m is the missile velocity. Dynamic pressure thus depends on two quantities which can be sensed or estimated over the duration of the missile's flight.

To obtain a lateral acceleration of the missile airframe, a force normal to the missile body axis must be applied. These forces are generated by either the deflection of an aerodynamic control surface or the operation of thrust vector control devices.

Most tactical missiles are controlled by means of aerodynamic control surfaces. These surfaces can be located forward on the missile body, a design called canard control, at the rear of the missile body, a design called tail control, or near the missile's center of gravity, as in conventional wing control. Canard control provides a fast speed of response and a high maneuverability. Tail control is preferred when roll attitude stabilization or roll control is required. Wing control provides a fast speed of response with low body rotation and a low angle of incidence. Combinations of these methods are also used. For example, canard control may be used for pitch and yaw, with tail-mounted roll control tabs.

Thrust vector control is applied to missiles operating in conditions of low dynamic pressure, which may result from low airspeed or high altitude. The lateral accelerations induced by thrust vector control are independent of dynamic pressure. Thrust vector control has been applied to guided weapons which require control almost immediately after launch.

The control of the motion of a tactical guided missile involves the control of a multi-variable dynamic system having at least two control inputs for motion in the longitudinal and lateral directions. There are many possible control system configurations which can be implemented to stabilize the behavior of this system and obtain a satisfactory transient response. The most common method for controlling a dynamic system is the use of feedback.

The control system designer must choose a basic structure for the feedback configuration. There are two major factors which affect this choice. First, control should be obtained by the application of the least control effort possible. A minimum control effort solution is desired because the use of unusually high feedback gains can result in motions which introduce flexing or bending responses in the missile body, and these unwanted motions can introduce nonlinearity and elasticity into the control system. These nonlinear effects are generally neglected at the time an initial mathematical model of the missile aerodynamics is constructed. A second consideration involves the simplicity of the feedback control system. Requirements for redundancy, safety, and failure detection will impose additional requirements on the control system which will eventually increase cost and complexity.

14.11 Future of PGMs

The future direction of technology in PGMs is to develop the brilliant munition capabilities indicated in Figure 14-10. The characteristics desired for brilliant munitions are detection, recognition, and identification. A brilliant munition must look over the battlefield and pick out a suspected target. Subsequent to this step, the brilliant munition must recognize what it has detected. Is the object a tank, truck, transporter-erector-launcher (TEL), a building, or a structure? The munition must look and recognize. Next, the brilliant munition must identify what it has detected and recognized. Is the object friend, foe, or neutral? Is it the highest value target? Is it the one that is assigned to be killed? The munition must look and identify. The final step is implied. The brilliant

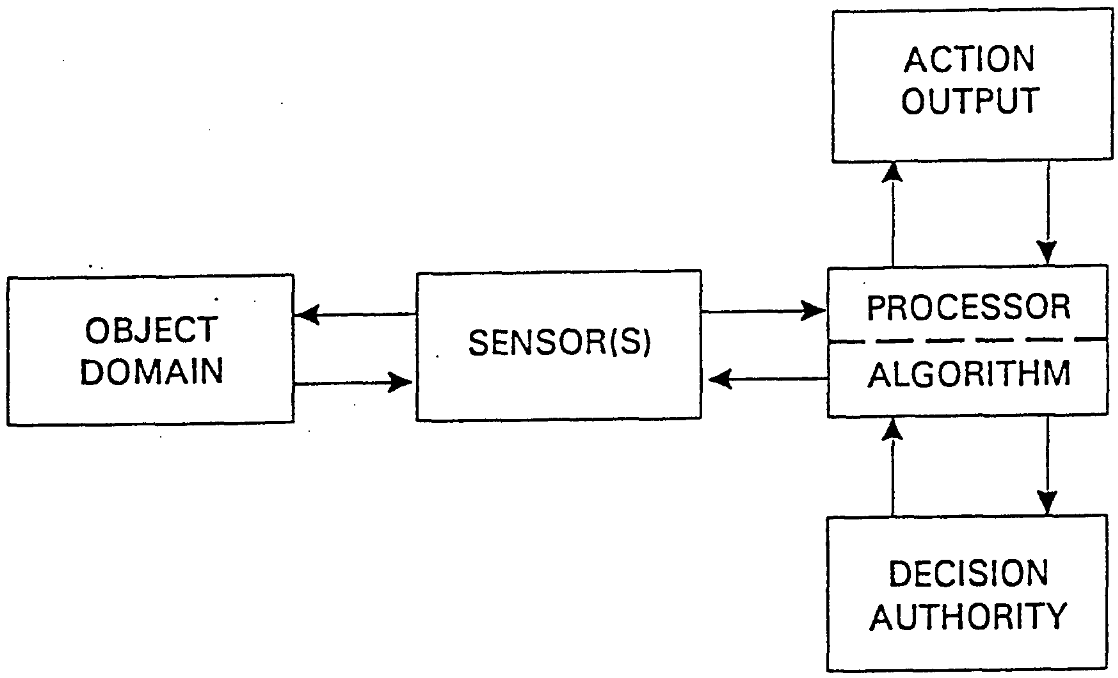


Figure 14-10. Generic ATR components.

munition must select an aim point. The munition must look and lock on. All of these steps must be performed without a man in the loop. Another important trait is that these steps must be performed with a minimum number of false alarms. Brilliant munitions must not be mistaken in their choices.

The primary advancement in technology that will yield brilliant munitions is automatic target recognition (ATR)[14.5]. The primary output of an ATR system is the characterization of an image of an object in anticipation of using this information to perform some action. When considered from this perspective, ATR technology has a high potential of dual-use applications. Military applications include target recognition by mines, submunitions, missiles, projectiles, and torpedoes. Space applications adapted ATR technology in the Brilliant Eyes and Brilliant Pebbles concepts. Reconnaissance, surveillance, target acquisition, and fire control systems can also make use of ATR. Aided target recognition, another interpretation of ATR, may be used to cue warfighters about potential targets. Dual-use options include remote sensors, terrain board imagery, unmanned vehicles (air, ground, and underwater), robotic sentries, computer vision, and face recognition. Mainly commercial uses include machine manufacturing robots, object recognition, and visual pattern recognition. The world of multimedia may open up other applications.

Figure 14-10 displays a generic ATR system. One or more sensors are used to view objects/targets with the sensory output being processed to cue a man/woman/machine to perform some action. Many different sensors have been considered singly or in combination for ATRs:

FLIRs, lasers, ladars, millimeter waves, synthetic aperture radars (SARs), and acoustics. The sensor of choice is dependent upon the desired application.

The processor in Figure 14-10 is a critical ATR hardware component. It should be small and lightweight for brilliant munitions and require as little power as possible (hundreds of watts). Massively parallel operations are required at hundreds of MIPS (millions of instructions per second) and thousands of MFLOPS (millions of floating point operations per second). The processor should also be easily reprogrammable, be trainable, handle higher order languages, and possess algorithm flexibility.

The algorithm block in Figure 14-10 is the software used by the processor to manipulate the images and signals coming out of the sensors so that a decision can be made about what is contained in the object domain. Algorithms are designed to analyze images of the object domain. Various operations are built into algorithms: feature extraction, edge detection, corner detection, segmentation, contour generation, 2-D and 3-D imaging, silhouette creation, model vision, or foveal vision. Feature extractors and segmentors seem to predominate. Multi-sensor fusion processes must be built into the algorithm. Countermeasures and programmability are key algorithm issues.

The decision authority in Figure 14-10 is a major aspect of the algorithm for ATRs. It is identified as a separate block because this is the primary function of an ATR—to recognize. The recognition process may be done in many ways: optical correlation, graph matching, neural net selection, template matching, statistical methods, binary tree, or model-based. Optical correlation and model-based recognition appear to be the current favorites.

Major progress has been made in advancing ATR technology over the past decade. There are still challenges. Target shadows and occlusions by trees and foliage are still a problem. A target-rich environment creates difficulties in choosing selected target types. It would be desirable to take minimal looks, even single scans, to identify targets and provide maximum time to pick a choice target. Programmability and trainability have already been mentioned, but are worth repeating. As ATR technology progresses, so will the potential for fielding brilliant munitions.

14.12 Summary

This chapter reviewed some of the basic classifications and components of Precision Guided Munitions (PGMs). Precision guided munitions were classified by launcher-target locations, target type, sensor operation, munition type, formal military designations, capabilities, range, direct guidance, and guidance laws. The future direction of PGMs is to adopt the capabilities offered by automatic target recognition to create a class of brilliant munitions.

REFERENCES

14.1 Heaston, R.J. and Smoots, C.W., Introduction to Precision Guided Munitions: A Handbook Providing Tutorial Information Data on Precision Guided Munitions (PGMs), Vol. 1, GACIAC HB-83-01, GACIAC/IIT Research Institute, Chicago, Illinois, May 1983.

14.2 DoD Directive 4120.15, Designating and Naming Military Aircraft, Rockets, and Guided Missiles, November 24, 1971. Cancelled and replaced by DoD Directive 4120.15, Designating and Naming Military Aerospace Vehicles, May 2, 1985.

14.3 Smart Munitions: An Introduction to the Concepts, The Technologies and the Systems, GACIAC SR-87-08, GACIAC/IIT Research Institute, Chicago, Illinois, November 15, 1987.

14.4 Garnell, P., and East, D.J., Guided Weapon Control Systems, Pergamon Press, New York, 1977.

14.5 Proceedings of the Third Automatic Target Recognizer Systems and Technology Conference, 29 June-1 July 1993, Vol.I (Unclassified/Export Control) and Vol.II (Secret-NOFORN).

CHAPTER 15
APPLICATIONS OF CONTROL THEORY TO PGMs

15.1 PGM Simulation

The mathematical model for the motion of a precision guided munition (PGM) having six degrees of freedom is a set of highly nonlinear strongly coupled differential and algebraic equations. These equations relate a large number of state variables and other factors to aerodynamic, gravitational, propulsion, and inertial forces and moments which act on a missile body during flight. Some of these forces and moments are controllable in the sense that they can be manipulated by an operator or a control system. Other forces and moments are not controllable and must be treated as disturbance inputs or noise sources. A number of approaches for disturbance accommodating control theory are presented in this chapter.

The complexity of the state variable model for a specific missile depends on the missile's physical design and the configuration of its control surfaces and on the flight regime in terms of altitude and Mach number. The derivation of the equations of motion for an aerodynamic vehicle have been detailed [15.1, 15.2].

To develop a complete mathematical model for the purposes of missile design and simulation it will also be necessary to develop mathematical models of the target, background, seeker, guidance computer, autopilot, actuators, propulsion system, and geometry of the engagement.

To model the geometry, several right-hand cartesian coordinate systems are required. The necessary systems include a fixed inertial system and systems attached to the target, seeker, and missile body. It will be necessary to develop the equations for transferring vectors from each of these coordinate systems to the other.

Digital computer simulations are now widely used by control system designers to develop solutions to design and development problems of tactical guided missiles and other weapons. Digital simulation has proven to be a useful and valuable tool for this work because it offers a safe, cost-effective means to evaluate the effects of design changes without the necessity of building and flying a hardware version of the system under study. There has been an explosive growth in modeling and simulation with the advancements in computers and software.

Once a complete digital computer model of a weapon system has been developed, programmed, tested, and been found to demonstrate the desired performance, certain sub-systems of the missile such as the seeker, guidance computer, or autopilot can be included in hardware form. A hardware-in-the-loop simulation of a missile system is thus developed by removing the software module that performs the same task as the hardware, providing the digital computer with a set of analog and digital inputs and outputs which allow signals to be exchanged between the digital computer program and the external hardware device, and executing the digital computer simulation in real-time to simulate the flight of the missile.

The stability, maneuverability, speed, and miss distance between the missile and its target are all of interest to the control system designer. These factors are ultimately determined by the aerodynamic forces and moments experienced by the missile during its flight. The forces and moments are computed using data derived from past experience, wind tunnel tests, or computations of the complex aerodynamic flows surrounding the missile body and control surfaces.

Once the forces and moments are known, usually as functions of flight parameters (including Mach number, altitude, the acceleration of gravity, the angles of attack and sideslip, and the control surface deflections), application of classical mechanical principles allows the translational and rotational accelerations to be computed. Integration of these accelerations then produces the translational and rotational velocities and displacements. The angular velocities include the pitch, yaw, and roll rates.

The development of the mathematical models for these accelerations requires the definition of a reference coordinate system. The forces and moments which drive the translational and rotational accelerations must be defined in terms of this reference system. For an aerodynamic missile the usual convention is to define a right-hand rectangular coordinate system having its origin at the missile's center of gravity, with the x-axis oriented forward along the axial length of the missile and the y-axis oriented in the direction of the right wing. Thrust and drag are then resolved into axial forces along the x-axis, side forces along the y-axis, and normal (lift) forces along the z-axis which generally points downward.

The roll, pitch, and yaw moments are then measured along the x, y, and z axes. The definition of this coordinate system allows the basic equations for a six degree of freedom mathematical model of the missile to be written. The six degrees of freedom refer to the three translational motions along the x, y, and z axes and the three rotational motions around the x, y, and z axes. The roll, pitch, and yaw angles are referred to as the Euler angles. For an aerodynamic missile having a single plane of geometric symmetry, the equations for the pitch, yaw, and roll angular accelerations are:

$$\frac{dP}{dt} = \frac{1}{I_{xx}}\left[\left(\frac{dR}{dt} + QP\right) I_{xz} - RQ\,(I_{zz} - I_{yy}) + L\right]$$

$$\frac{dQ}{dt} = \frac{1}{I_{yy}}\left(-\left(P^2 - R^2\right) I_{xz} - PR\,(I_{xx} - I_{zz}) + M\right)$$

$$\frac{dR}{dt} = \frac{1}{I_{zz}}\left[\left(\frac{dP}{dt} - QR\right) I_{xz} - PQ\,(I_{yy} - I_{xx}) + N\right]$$

where

P = P(t) = roll rate, radians per second,

Q = Q(t) = pitch rate, radians per second,

R = R(t) = yaw rate, radians per second,

I_{xx} = moment of inertia about the x-axis,

I_{yy} = moment of inertia about the y-axis,

I_{zz} = moment of inertia about the z-axis,

I_{xz} = product of inertia,

L = applied roll moment,

M = applied pitch moment,

N = applied yaw moment.

The lateral accelerations can be written as:

$$\frac{du}{dt} = \frac{1}{m}\sum F_{xb} - Qw + Rv,$$

$$\frac{dv}{dt} = \frac{1}{m}\sum F_{yb} - Ru + Pw,$$

$$\frac{dw}{dt} = \frac{1}{m}\sum F_{zb} - Pv + Qu,$$

where

u = u(t) = velocity in the x-direction,

v = v(t) = velocity in the y-direction,

w = w(t) = velocity in the z-direction,

P, Q, and R are the roll, pitch, and yaw rates,

ΣF_{xb} = summation of the applied forces ϵ the x-direction,

ΣF_{yb} = summation of the applied forces ϵ the y-direction,

ΣF_{zb} = summation of the applied forces ϵ the z-direction,

m = the missile mass.

The time-varying applied forces in each direction are due to propulsion, lift, drag, and the influence of gravity. The time-varying applied moments are due to aerodynamic effects and are dependent on the missile altitude, dynamic pressure, control surface deflection, and possibly angle of attack and side angle.

For control purposes the time-varying missile velocities u, v, and w must be transformed from the missile body coordinate system into a ground reference system. One or more transformation matrices, which track the angular rotations of each of the coordinate systems involved and permit vectors in any one system to be transferred into another coordinate system, must also be updated throughout the missile's simulated flight.

The time-varying differential equations which comprise the basic six degree of freedom model of an aerodynamic missile are a set of highly nonlinear closely coupled equations which, in the form above, cannot be solved by means of LaPlace transform methods. Thus, classical methods of control system design, including the use of transfer function and frequency response methods, cannot be applied to these equations in their present form. By selecting an operating point for the missile in terms of an angle of attack or pitch angle, missile velocity, and control surface deflection, and assuming that the angular rates will be maintained very near to zero, a set of linearized equations can be obtained.

15.2 Theory of Disturbance Accommodating Controllers

All realistic control systems operate in environments which produce system disturbances. These disturbances are treated as system inputs which cannot be accurately predicted and are uncontrollable in the sense that they cannot be controlled by the system designer. Disturbances usually introduce unwanted disruptions into the otherwise orderly behavior of a controlled dynamic system. Examples of disturbances encountered in practice include:

(a) uncertain fluctuating loads on speed regulators and power generators,

(b) uncertain flow and reaction rates in chemical processes,

(c) wind gusts, updrafts and other time-varying aerodynamic effects,

(d) friction, center-of-gravity offsets, thrust misalignments and other uncertain bias effects in mechanical and electrical systems.

A properly designed control system must effectively cope with a range of disturbances foreseen by the system designer. A high-performance control system should be designed so as to maintain the given control system design specifications in the face of all disturbances that might act on the dynamic system under actual operation conditions.

Classical control system design methodology includes clever and highly-effective methods for dealing with step, ramp, and sinusoidal disturbances in simple single-input, single-output time-invariant control systems. These methods are largely heuristic in nature and include the use of integral control action, feedforward control action, and notch filters to modify the steady-state error characteristics of the dynamic system's closed-loop transfer function.

The block diagram in Figure 15-1 shows a speed control system[15.3] in which the output is subject to a torque disturbance whose Laplace transform is N(s). R(s) represents the reference speed input, or setpoint, C(s) the output angular velocity of a rotating member, E(s) the error between the input and the output, T(s) the applier torque produced by the control system, J is the rotational moment of inertia of the rotating member, and K is the gain of the control system. When no disturbance is present, the output speed equals the input speed.

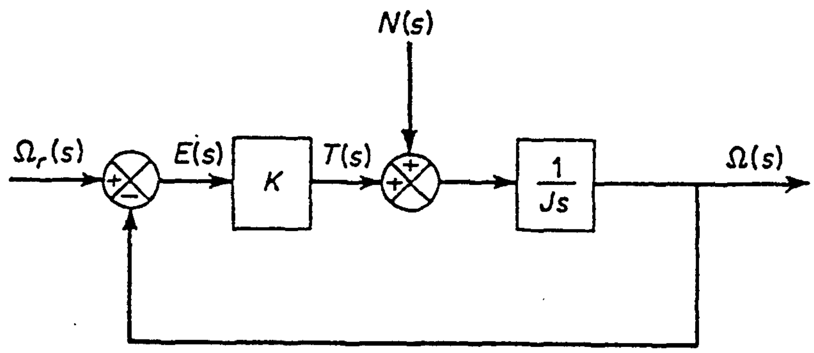


Figure 15-1. Block Diagram of a Speed Control System

The response of this linear time-invariant system to a unit step torque disturbance can be examined by forming the transfer function between N(s) and C(s), assuming that the reference input R(s) equals zero:

$$\frac{C(s)}{N(s)} = \frac{1}{(J_s + K)} .$$

The steady-state error in response to a unit step disturbance input can be obtained by applying the final value theorem:

$$c(@@) = \lim s \geq 0 \left[\frac{s}{(J_s + K)} \right] * \left[\frac{1}{s} \right] = \frac{1}{K}.$$

If a unit step disturbance is applied to the system, a speed error equal to 1/K will result. The resultant applied torque will cancel the effect of the disturbance torque, so that the rotation eventually reaches a steady state condition, but the final speed will no longer equal the reference speed. Note that the response due to a change in setpoint may be added to the response due to a disturbance torque, since the dynamic system is linear and time-invariant.

By modifying the control system it is possible to cancel the effect of a disturbance torque at steady-state so that a constant disturbance torque will cause no speed change at steady-state. A controller whose transfer function is Gc(s) must be designed. Ignoring the reference input, the resulting transfer function between the disturbance torque and the output is:

$$\frac{C(s)}{N(s)} = \frac{1}{(J_s + G_c(s))}.$$

The steady-state error in response to a unit step disturbance input can again be obtained by applying the final value theorem:

$$c(@@) = \lim s \geq 0 \left[\frac{s}{(J_s + G_c(s))} \right] * \left[\frac{1}{s} \right] = \frac{1}{G_c(0)}.$$

To obtain a final value of c(@@) equal to zero, the value of Gc(O) must be infinite. This can be provided by a controller having the transfer function:

$$G_c(s) = \frac{K}{s}.$$

This controller consists of an integrator with a gain of K. The resulting integral action will continue to correct the output speed in response to a unit step disturbance torque input until the steady-state output speed is zero. The use of this controller causes other problems, however. The transfer function of the resultant closed-loop system is now:

$$\frac{C(s)}{R(s)} = \frac{\left[\frac{K}{J}\right]}{\left[S_2 + \left[\frac{K}{J}\right]\right]}.$$

The closed-loop system is now unstable. The transfer function now has a pair of complex poles located on the jw-axis at $\pm$(K/J)1/2. The steady-state response to a simple unit-step change in the setpoint is now a continuous sinusoidal oscillation, rather than a constant output speed.

The system can be stabilized by the addition of a proportional mode controller which results in the following controller transfer function:

$$G_c(s) = K_p + \frac{K}{s} = \frac{(K_p s + K)}{s}.$$

When this controller is implemented, the transfer function between the disturbance torque and the output becomes:

$$\frac{C(s)}{N(s)} = \frac{s}{\left(J_s^2 + K_p s + K\right)}.$$

The steady-state error in response to a unit step disturbance input can again be obtained by applying the final value theorem:

$$c(@@) = \lim s \geq 0 \left[\frac{s}{(J_s + G_c(s))}\right] * \left[\frac{1}{s}\right] = \frac{1}{G_c(0)}.$$

The proportional-plus-integral controller thus eliminates any speed error at steady-state due to the presence of a unit step disturbance step input. The response of the closed-loop system to a unit-step change in the reference input can be obtained using the closed-loop transfer function:

$$\frac{C(s)}{R(s)} = \frac{(K_p s + K)}{\left(J_s^2 + K_p s + K\right)}.$$

This transfer function has a pair of poles located at:

$$s_{1,2} = \frac{[-K_p \pm \left(K_p^2 - 4KJ\right)^{1/2}]}{[2J]}$$

The response to a unit step change in the setpoint will depend on the location of these poles in the complex plane. Normally, a damping factor of about 0.7 will be selected, resulting in a damped oscillatory response which settles in a reasonably short time to a final value equal to the new setpoint.

With this design, a step disturbance torque will produce a transient error in the output rotational speed, but the error will become zero under steady-state conditions. The integrator provides a nonzero output even when the error reaches zero. This output produces a motor torque which exactly cancels the effect of the unit-step disturbance torque.

Modern control theory allows more complex problems having many inputs and many outputs to be addressed and investigated using powerful mathematical methods based on matrix algebra. Modern control technology has been somewhat slow to address the fundamental problem of how to deal with unwanted disturbances in multivariable control systems. Virtually all early papers on modern control theory and almost all textbooks currently dealing with the design of multivariable control systems deal with the following mathematical model for a linear dynamic system:

$$\frac{dx(t)}{dt} = A(t)\,x(t) + B(t)\,u(t),$$
$$y(t) = C(t)\,x(t) + D(t)\,u(t)$$

In this model the state variables are represented by the vector x(t), the output by the vector y(t), and the only allowable inputs by the vector u(t). Feedback control laws can be derived using methods from optimal control theory which regulate the state of this system about some nominal state, but these control laws are of the form u(t) = u(x(t), t). Such control laws cannot effectively control a multivariable system confronted by unknown disturbances.

The method of disturbance accommodating controllers, has been refined and developed as a practical, general purpose design tool suited for a wide variety of multivariable control system applications. In this report the focus is on continuous-time control systems, but the disturbance accommodating method has also been extended to discrete time control systems. This method provides the control system designer with a systematic method for designing multivariable feedback control systems which are effective in coping with those kinds of persistent disturbances encountered in practical applications. Although applicable to a wide class of multivariable control systems, design specifications and disturbances than classical control theory, the disturbance accommodating method automatically produces the classical control system designs (integral control action, feedforward control and notch filters) when the dynamic systems, design specifications and disturbances considered are reduced to the single-input, single-output case.

Disturbance accommodating control theory can be viewed as a modern control theory state variable implementation of the traditional integral, feedforward, and notch filter control methods historically proven effective in classical control system design.

15.2.1 Waveform Mode Description of Realistic Disturbances

Disturbances encountered by practical control systems can be classified as either noise-type disturbances or disturbances with waveform structure. Noise-type disturbances exhibit time-recordings which are random, jagged, and erratic in nature, having no significant regularity or smoothness. Examples of these disturbances are radio static, motor brush noise, and fluid turbulence. In contrast, time-recordings of disturbances having waveform structure exhibit recognizable waveform patterns, at least over short recording time intervals.

Noise-type disturbances, which have no waveform structure, are traditionally characterized by their statistical properties (mean value, covariance, power spectral density, etc.). Noise-type disturbances are traditionally modeled in terms of random processes, white noise, and colored noise.

Stochastic stability, filtering, and control theory are concerned almost exclusively with noise-type disturbances.

Disturbances which are classified as waveform structure disturbances exhibit readily distinguishable waveform patterns over short time intervals. Waveform structure disturbances can be modeled mathematically by semi-deterministic analytical expressions of the form:

$$w(t) = W\,(f_1(t), f_2(t), \ldots, f_m(t);\ c_1, c_2, \ldots, c_L),$$

where the $f_i(t)$ are known time functions and the c_L are unknown parameters which may occasionally jump in a random, piecewise constant manner from one value to another. A mathematical model of this sort is called a waveform-mode description of the disturbance. The collection of known functions $f_i(t)$ reflect the waveform patterns observed in the experimentally recorded data.

A waveform-mode description having special importance is the linear case:

$$w(t) = c_1 f_1(t) + c_2 f_2(t) + \ldots + c_m f_m(t).$$

The collection of time functions $f_i(t)$ forms a finite basis for the m-dimensional function space and the c_i are a set of piecewise constant weighting coefficients. The unknown disturbance can be expressed as a linear combination of these prescribed basis functions, each weighted by a coefficient c_i. The c_i may jump in value from time to time in a random piecewise constant manner. By inspection it can be determined that this disturbance is generally comprised of a sum of weighted linearly combined steps and ramps. This disturbance can be represented mathematically by:

$$w(t) = c_1 + c_2 t,$$

where the weighting coefficients c_1 and c_2 vary in a random piecewise constant manner. The two basis functions for this mathematical representation are:

$$f_1(t) = 1,\ \ f_2(t) = t.$$

Disturbances having these representations are known to occur as load fluctuations on speed and power regulators, temperature, pressure, and flow variations in chemical reactors, pulse, and shock inputs in electrical and mechanical systems, mechanical friction, etc.

The theory of disturbance accommodating control was developed to accommodate a broad class of realistic control system disturbances which can be defined in terms of a waveform-mode description. The theory provides a general design tool for developing control systems for dynamic systems affected by disturbances having waveform structure. This theory is comparable to the stochastic control theories available as design tools for the control of dynamic systems affected by statistically modeled noise-type disturbances.

15.2.2 Waveform-Mode Versus Statistical Representation of Disturbances

The characterization of unknown disturbances in terms of a waveform-mode description represents a significant departure from the traditional approach to disturbance modeling. The information contained in the waveform-mode description is much different than the information about the unknown disturbance contained in the traditional statistical measures (mean value, covariance, power spectral density, etc.) Since the time behavior of the weighting coefficients is assumed to be completely unknown except for the fact that the c_i vary in a random, piecewise continuous manner, the waveform-mode description does not provide any indication of the mean value, covariance and other common statistical measures of the disturbance.

Common statistical measures of disturbances such as mean value, covariance, and power spectral density are based on longterm averages. However, the disturbance information that is meaningful in the design of a high-performance control system is the short-term or current behavior of the disturbance. The short-term statistical properties of most unknown disturbances do not usually exist, since the short-term mean, covariance, etc., are themselves random variables. Consider as an example the problem faced by the driver of a small car traveling down a 20-mile stretch of highway and faced with a strong fluctuating crosswind. A good driver would steer the car in accordance with short-term, instantaneous behavior of the crosswind, the wind as it actually affects the behavior of the car. Advance knowledge of the statistical properties of the wind as measured over the entire 20-mile route would be of little or no use to the driver attempting to make on-line, real-time steering decisions. In a classical control system design approach, the statistical properties of the crosswind might be used to predict the average position of the steering wheel and the variance about that average position, measured over the entire 20-mile stretch of highway.

Effective on-line, real-time control of dynamic systems requires information about the short-term current behavior pattern of the actual disturbance sample function w(t). Long-term average statistical properties do not reveal the information required, and the desired short-term information does not typically lend itself to a meaningful statistical representation in classical terms. A control system design which relies on long-term statistical properties is justified only when the behavior pattern of the disturbance is erratic, jagged, and devoid of any waveform structure. In that case the disturbance is classified as noise and the use of stochastic control techniques and random process models is the best approach available to the designer.

The waveform-mode description was conceived as a means for filling the information gap attributable to the classical statistical characterization. The waveform-mode description describes the range of possible waveform shapes or structures that a particular unknown disturbance w(t) might exhibit at any point in time. From a random process viewpoint, the waveform-mode description in terms of a set of basis functions and a set of time-varying coefficients can be viewed as a mathematical model of the M-parameter family of sample-functions [w(t)] from which the actual disturbance w(t) is expected to be produced. The waveform-mode description is not a random-process sample function in the usual sense because the set of basis functions [$f_i(t)$] was not selected to match any statistical properties of w(t). The basis functions f(t) are selected by the control system designer to match the distinctive waveform shapes and behavior patterns observed by the designer in experimental recordings of w(t) taken under realistic conditions. Each individual sample function w(t) is permitted to have a different set of long-term statistical properties. The waveform-mode description is thus applicable to highly non-ergodic disturbances, including the commonly encountered situation in which each sample function w(t) has a different random but constant value (the disturbance is a step function of random amplitude).

If a control system designer can confidently represent an unknown disturbance affecting a dynamic system by a waveform-mode description then the designer can disregard all statistical considerations, random-process theories, stochastic control methodologies, etc., and can proceed to design a physically realizable deterministic-type feedback control system, a disturbance accommodating controller, which is effective in coping with the specified class of disturbances. When the disturbances have a waveform structure, a disturbance accommodating controller will yield significantly better performance than a stochastic controller designed by considering only long-term statistical properties of the disturbance.

15.2.3 State Variable Modeling of Waveform Structure

To develop a waveform-mode representation of a disturbance, the control system designer must first select a set of basis functions to represent the disturbance. This may be done by means of visual and computer-aided analysis and inspection of recorded experimental data, or by means of an analysis of the dynamic system which produces the disturbance.

Once a suitable set of basis functions has been selected, a mathematical model of the disturbance in state variable form must be constructed. This can be done in a number of ways. For example, if each of the basis functions has a Laplace transform:

$$L\,[f_i(t)] = f_i(s) = \frac{Pm_i(s)}{Qn_i(s)},$$

where the numerator polynomial $Pm_i(s)$ is of degree m_i and the denominator polynomial $Qn_i(s)$ is of degree n_i and $0 \le m_i \le n_i <$ @@, and if the coefficients c_i are temporarily treated as constants, the Laplace transform of the disturbance can be written as:

$$w(s) = L\big[w(t)\big] = \sum_{i=1}^{m} c_i\, f_i(s)$$

$$= \sum_{i=1}^{m} c_i \frac{Pm_i(s)}{Qn_i(s)}$$

By consolidating terms this can be written as the single ratio of two polynomials:

$$w(s) = \frac{P(s)}{Q(s)},$$

where the numerator polynomial involves the coefficients c_i and the denominator polynomial is the least common denominator polynomial of the set of denominator polynomials for the Laplace transforms of the basis functions.

The denominator polynomial Q(s) can be written as:

$$Q(s) = s_p + q_p s_p - 1 + q_p - 1 s_p - 2 + \ldots + q_s^2 + q^1,$$

where

$$p \le \sum_{i=1}^{m} n_i\,.$$

The disturbance s(t) can then be developed as the output of a fictitious linear dynamic system having the transfer function:

$$G(s) = \frac{1}{Q(s)},$$

and subject to a set of initial conditions [w(0), w'(0), w"(0). ...,] which, when Laplace transformed, yield the numerator polynomial P(s). The disturbance w(t) satisfies the linear time-invariant homogeneous differential equation:

$$\frac{d^p w}{dt^p} + \frac{q^p d^{(p-1)} w}{dt^{(p-1)}} + \ldots + \frac{q^2 dw}{dt} + q_w^1 = 0,$$

where the coefficients q_i are explicitly known since they are independent of the coefficients c_i and depend only on the assumed set of basis functions.

The constant coefficients c_i assumed in this development are, in reality, only piecewise constant. Their values may change from time to time in an unknown randomlike manner. To account for the random changes in the coefficients c_i an external forcing function w(t) is defined which consists of a sequence of completely unknown, randomly arriving, random intensity impulse functions (deltas, doublets, triplets, etc.). This yields a state variable model for the disturbance having the form:

$$\frac{d^p w}{dt^p} + \frac{q^p d^{p-1} w}{dt^{p-1}} + \ldots + \frac{q^2 dw}{dt} + q^1 w = w(t).$$

The forcing function w(t) is indeed completely unknown and is included in the model for the symbolic purpose of providing a mathematical reason for the jumping of the c_i in the model of the disturbance.

This single p^{th} -order differential equation can be written in state variable matrix format:

$$w\ (t) = z^1(t)\ ,$$

$$\frac{dz^1(t)}{dt} = z^2(t) + \sigma^1(t)\ ,$$

$$\frac{dz^2(t)}{dt} = z^3(t) + \sigma^2(t)\ ,$$

$$\ldots\ ,$$

$$\frac{dz^{(p-1)}(t)}{dt} = z^p(t) + \sigma^{(p-1)}(t)\ ,$$

$$\frac{dz^p(t)}{dt} = -\ q^1 z^1(t) - q^2 z^2(t) - \ldots - q^p z^p(t) + \sigma^p\ (t)\ ,$$

where the symbolic effect of w(t) in the differential equation is now represented by the completely unknown functions $\sigma(t)$ which are sequences of completely unknown, randomly arriving, random intensity impulse functions.

The arrival times of adjacent impulse functions, which model the adjacent jumps in the coefficients c_i, are assumed to be separated by some minimal time spacing $\mu > 0$. The smallest minimal acceptable value of μ will depend on the response times of the controller hardware in any specific application.

More complicated models for unknown disturbances may involve time-varying coefficients q_i or nonlinear terms involving w(t), dw(t)/dt, etc. The desired state variable model for such a process may be either a single nonlinear time-varying differential equation or a set of nonlinear time-varying first-order differential equations. If w(t) is a multivariable disturbance having p components:

$$w(t) = \left(w_1(t),\ w_2(t),\ \ldots,\ w_p(t)\right)\ ,$$

then a separate state variable model must be derived for each independent component.

The model which is arrived at as a mathematical representation of the unknown disturbance has the same state variable form as the mathematical model used to describe the dynamic system which is to be controlled. The p-dimensional vector w(t) can thus be referred to as the state of the disturbance. The state of a disturbance is a fictitious quantity, a mathematical artifice which results from the modeling process. This contrasts with the state of a dynamic system, a quantity always related to some physical quantity in the dynamic system. It can be shown that the value of the instantaneous state z(t) of an uncertain disturbance w(t) represents all the information needed to make a rational, scientific, on-line decision for the applied control action u(t) at time t, even though the future behavior of the disturbance is unknown. This result is called the principle of optimal disturbance accommodation.

The numerical value of the order p of the differential equation representing the uncertain disturbance depends on the designer's choice for the basis functions $f_i(t)$. Basis functions chosen to fit the recorded data over longer time intervals generally result in smaller values for both m and p, at the expense of making the algebraic structure of the state variable somewhat more complicated. If the basis functions are selected to produce a good fit over relatively short time intervals, the state variable structure is somewhat simplified, at the expense of generally larger values for m and p. This latter selection can result in a disturbance model in which more frequent jumps in the numerical values of the c_i occur. This can decrease the performance of a disturbance accommodating controller.

As an example of the process of constructing a state variable model for an uncertain disturbance, consider a step/ramp disturbance. The waveform-mode description for this disturbance is:

$$w(t) = c_1 + c_2 t \ ,$$

and w(t) satisfies the second-order differential equation:

$$\frac{d^2 w(t)}{(dt^2)} = w(t) \ ,$$

where the term w(t) represents a completely unknown sequence of randomly arriving, random intensity impulses and doublets.

An equivalent representation of this uncertain disturbance can be written in terms of an output equation and a pair of first-order linear differential equations:

$$w(t) = [1,\ 0] \left[z_1(t)\ z_2(t)\right]^T \ ,$$

$$\frac{d}{dt\,[z_1(t)]} = [0,\ 1]\,[z_1(t)] + [\sigma_1(t)]$$

$$[z_2(t)]\ [0,\ 0]\ [z_2(t)]\ [\sigma_2(t)]\ ,$$

where the terms $\sigma_1(t)$ and $\sigma_2(t)$ represent two completely unknown sequences of randomly arriving impulse functions of unknown intensity.

Note that this process can be simulated by a linear system constructed from two integrators and the necessary interconnections. At random times separated by at least the minimum amount, the initial condition on one of the two integrators is changed to a randomly selected value. The integrator whose initial condition is changed is also selected at random. To approximate a process in which the interarrival times are truly independent, a modified Poisson process in which the interarrival times are exponentially distributed can be used. The modification consists in assigning a finite probability to the small but finite minimum allowable interarrival time.

15.2.4 A Waveform Description of Unfamiliar Disturbances

In the absence of reliable test data illustrating the nature of the disturbance acting on the dynamic system, the control system designer can proceed by assuming that the disturbance is modelled by a polynomial of the form:

$$w(t) = c^1 + c^2t + c^3t^2 + \ldots + c^pt^{p-1}.$$

This represents the uncertain disturbance as a power series over time. This is claimed to be effective for unfamiliar disturbances which vary rather slowly.

For unfamiliar disturbances which arise as a result of modelling errors or variations in the parameters of the dynamic system a waveform description can also be used. For example, if the dynamic system is believed to be represented by a state-variable model having the following form:

$$dx(t)/dt = A\,x(t) + B\,u(t),$$

but that due to modelling errors, parameter drifts or other sources of variation the actual system behaves according to:

$$dx(t)/dt = (A + dA)\,x(t) + (B + dB)\,u(t),$$

the model of the system can be rewritten as:

$$dx(t)/dt = A\,x(t) + B\,u(t) + (dA\,x(t) + dB\,u(t)),$$

and this can in turn be modelled by:

$$dx(t)/dt = A\,x(t) + B\,u(t) + w(t).$$

The added disturbance w(t) can now be interpreted as including all the parameter mismatch terms as well as any unknown disturbance present in the system. A waveform description of w(t) can be used in a disturbance accommodating controller. This model can be used as a device for adaptive control of a system with time-varying parameters.

Attention may be focused on waveform descriptions based on linear state-variable models, rather than higher-order differential equations or nonlinear state-dependent disturbance models. The reasons for this focus are that satisfactory results are often obtained using these simpler models, the mathematical theory for the resulting differential equations is well-developed, and often it is necessary to linearize a nonlinear dynamic system model about some operating point. The control system designer's attention can thus be restricted to waveform descriptions having the form:

$$w(t) = H(t)\, z(t),$$

$$dz(t)/dt = D(t)\, z(t) + \text{sigma}(t),$$

or the state-dependent form:

$$w(t) = H(t)\, z(t) + L(t)\, x(t),$$

$$dz(t)/dt = D(t)\, z(t) + M(t)\, x(t) + \text{sigma}(t),$$

where H(t), D(t), L(t) and M(t) are matrices which are assumed to be known once the model of the disturbance has been developed.

Almost any realistic disturbance w(t) likely to be encountered in practice can be modelled by one of these linear forms, including combinations of constants, steps, ramps, accelerations, and general polynomials of time, decaying or growing exponentials, decaying, growing or steady-state sinusoids, sequences of pulses, oscillations with time-varying frequency, exponentials with time-varying time constants, polynomials with fractional powers and any other function which satisfies a linear, time-varying or time invariant differential equation.

15.2.5 The Design of Disturbance Accommodating Controllers for Stabilization, Regulation and Tracking Control Problems

The waveform-mode description of unknown disturbances can be combined with methods drawn from the state-variable analysis of dynamic systems to produce a variety of high-performance feedback controllers called disturbance accommodating controllers. These controllers yield high-quality closed-loop performance when the closed-loop system is confronted with a wide range of transient and persistent disturbances.

The controlled system is modelled by a set of linearized state-variable equations having the form:

$$dx(t)/dt = A(t)\,x(t) + B(t)\,u(t) + F(t)\,w(t),$$
$$y(t) = C(t)\,x(t) + E(t)\,u(t) + G(t)\,w(t),$$

where x is the system state vector, u is the system input vector, w is the uncertain disturbance vector and y is the output vector. The time-varying matrices A(t), B(t), F(t), C(t), E(t), and G(t) are assumed to be known, and may be constant.

The uncertain disturbance w(t) is modelled by a linearized waveform-mode description:

$$w(t) = H(t)\,z(t) + L(t)\,x(t),$$
$$dz(t)/dt = D(t)\,z(t) + M(t)\,x(t) + \text{sigma}(t),$$

where the vector z is interpreted as the state of the disturbance w. The time-varying matrices H(t), L(t), D(t), and M(t) are also assumed to be known.

In a realistic control system design it is usually impractical to perform direct on-line measurement of all the system state variables. Similarly, direct, on-line measurements of all the disturbance components are also impractical and often impossible. In the design of a disturbance accommodating controller it is thus assumed that the controller is allowed to operate based on only three quantities:

a) The real-time, on-line measurements of the system output y(t),

b) The real-time, on-line values of the reference inputs, set points and control actions, and,

c) The particular subset of disturbance components $w_i(t)$ which can be physically measured in an on-line, real-time manner. In practice it may turn out that none of the disturbance components can be measured. This will not prove to be a problem for the designer.

Control problems generally fall naturally into one of three categories: stabilization problems, regulation problems or tracking problems. The stabilization problem involves the design of a feedback controller which will cause the system state to return to and stay at an equilibrium value, typically the state-space origin, in the face of initial state perturbations or external disturbances.

Regulation problems are similar to stabilization problems, but the system state is required to consistently return to a reference state determined by a set of setpoints or reference inputs. Initial errors between the system state and the reference state may be present, as well as external disturbances which require the corrective action of a closed-loop control system. The reference inputs may be changed from time to time and are not necessarily equilibrium points of the dynamic system.

Tracking or servomechanism problems involve the design of a closed-loop controller which will cause the system state x(t) to consistently follow or track a time-varying command input. The command input may or may not be known in advance, but can usually be measured on-line. The command input must be followed consistently even in the face of external disturbances or initial errors. The tracking problem is the most general of these three, and includes the stabilization and regulation problems as special cases.

Modern control theory, specifically the application of optimal control theory, allows a designer to develop a feedback controller for a linear, time-varying dynamic system that performs stabilization, regulation or tracking while simultaneously minimizing some important performance measure such as the integral square error, expended control energy, or the transition time from an initial state to a specified final state.

A system designer attempting to cope with the presence of external, unknown disturbances which affect the performance of a control system can adopt one of three attitudes regarding the disturbance. A common presumption is that the disturbance is always undesirable and degrades the performance of the closed-loop control system. For that case, a disturbance absorbing controller can be designed which optimally accommodates the disturbance by exactly canceling out all effects of the disturbance on the behavior of the system, thus eliminating the effect of the disturbance.

In some cases it may be impossible for the controller to exactly cancel all of the disturbance's effects. Then the designer must adopt the attitude that the disturbances are optimally accommodated when the controller is designed to minimize the effects of the disturbance rather that totally eliminate them. A controller designed based on this approach is called a disturbance minimizing controller. The design of this controller will depend on the specific disturbance effect minimized by the control system designer.

If the disturbance might occasionally be capable of producing a desirable effect on the behavior of the dynamic system, the disturbance will be optimally accommodated when the controller is designed in such a way as to harness and exploit all potentially useful effects produced by the disturbance. For example, the disturbance might assist the controller in steering the system state from an initial state to some desired final state. A controller which performs this function will be called a disturbance utilizing controller.

These three design concepts might be combined in the design of a multi-mode disturbance accommodating controller. For example, a regulator design might employ a disturbance utilizing mode when large system errors are present and a disturbance absorbing controller after the error has been reduced to a sufficiently small value.

The design of a disturbance accommodating controller for a linear time varying system leads to a control law given by an expression of the form:

$$u(t) = U(x(t), z(t), t),$$

where x(t) is the system state vector and z(t) is the disturbance state vector. Neither the system state nor the disturbance state are usually available for direct measurement. The only quantities normally available to the control system designer are the output y(t), the setpoint or reference inputs, and perhaps some of the disturbance components. The designer can generate the required on-line data x(t) and z(t) based upon on-line measurements of available data by employing a state observer for signals with waveform structure. State observers are also called state constructors.

15.2.6 State Observers for Signals With Waveform Structure

The instantaneous state of an undisturbed linear dynamic system described by the mathematical model:

$$dx(t)/dt = A(t)\,x(t) + B(t)\,u(t),$$

$$y(t) = C(t)\,x(t) + E(t)\,u(t)$$

can be estimated by means of a state observer, a special purpose processor which operates only on the system output y(t) and the control action u(t).

If the uncertain disturbance w(t) has a waveform structure modelled by a set of linear, possibly time-varying, state-variable equations, then it is also possible to design and implement an observer which will provide a reliable, accurate, on-line estimate of the state of the disturbance. The disturbance state estimator will operate only on the output of the system, y(t), the control input u(t), and those components of the disturbance which might be measurable. By consolidating into a single mathematical model the state variable equations for the dynamic system and the uncertain disturbance, a composite state observer can be designed which produces all the data needed to compute the required feedback control actions.

A composite state observer can be used to develop and implement feedback controllers of the form:

$$u(t) = U(x'(t), z'(t), t),$$

where x'(t) and z'(t) are on-line estimates of the state of the dynamic system and the disturbance. If the estimation errors $e^{x(t)} = x(t) - x'(t)$ and $e^{z(t)} = z(t) - z'(t)$ approach zero quickly as compared to other settling times in the dynamic system, the feedback controller which results will do a good job of control and the resulting closed-loop system will be a good engineering approximation to the ideal closed-loop disturbance accommodating controller.

15.3 Closed-Loop System Analysis Using Lyapunov Stability

The second method of Lyapunov, discussed in Chapter 3 of this review, provides a useful approach for determining the stability of a dynamic system. This method, also called the direct method, involves the selection of a generalized scalar potential function, called a Lyapunov function. The selected Lyapunov function is tested to determine if it meets certain technical conditions which indicate the stability of the underlying dynamic system. Lyapunov functions are not unique for any specific dynamic system, and may be difficult to develop for some dynamic systems.

The second method of Lyapunov provides only a sufficiency test for stability. This means that if the selected Lyapunov function does not meet the test criterion, the underlying dynamic system may still be stable. In that case, if it is suspected that the underlying dynamic system is stable and if that stability must be demonstrated, a different Lyapunov function must be selected. Once a Lyapunov function indicating stability is found, that function provides a tool for further relative stability analysis and control system design.

The application of Lyapunov's second method to the stability analysis of a closed-loop control system containing a state variable observer in the feedback loop has not received much attention in the past. Geering and Baser[15.4] identified a Lyapunov function for the linear quadratic regulator problem with a full-order linear state observer.

In their work, Geering and Baser identified a Lyapunov function for the linear quadratic regulator problem and used the solution provided in a performance measure of the form:

$$J = q^T V q$$

where

$q = [x^T, e^T]^T$,

x = the true state vector, and

e = the observer error.

Geering and Baser were able to show, by means of a Lyapunov function, that the linear quadratic regulator problem has a superior control gain, and superior performance, for every arbitrary choice of the observer gain, only if the observer, which is itself a dynamic system, is initialized with the true values of the state variables.

The stability analysis of arbitrary closed-loop control systems containing state variable observers is important for several reasons. First, in a realistic environment, the complete and true state variable information will not be available for use in implementing the control law. This may be due to the physical unobservability of certain states, or the expense incurred to implement their measurement.

The certainty equivalence principle is the basis for designing most closed-loop control systems containing a state variable estimator. In these systems the optimal controller is designed separately, ignoring the unavailability of state variables or additive noise present in the measured state variables. A state variable estimator is then designed in the form of a Kalman filter, which provides estimates of the state variable values, and the two are combined in cascade to implement closed-loop control.

15.3.1 Homing Missile Guidance With Angle-Only Measurements

Optimal control theory and optimal estimation theory have increasingly been applied to the problem of homing missile guidance. The method of control most commonly used is based on linear-quadratic-Gaussian theory, which requires the use of a linear system model but yields a closed-form solution for the closed-loop control system.

A fundamental limitation to the application of linear-quadratic-Gaussian theory is the requirement to obtain accurate measurements of all the state variables of the dynamic system. In the homing missile application the linear-quadratic-Gaussian control law which comprises the guidance computation requires full knowledge of missile-to-target position and velocity and target acceleration. A modern missile can measure its own acceleration via on-board accelerometers. When a passive seeker such as an IR sensor is used, a measure of line-of-sight angle and rate can be developed. The full information required to implement the optimal control law is thus not directly obtainable, and some form of estimation process is thus required.

The primary focus of Vergez'[15.5] work was the development of a means for analyzing the performance of a closed-loop control system with a state variable observer in the feedback loop. The observer provided those estimates of the dynamic system's state variables required to implement a closed-loop control law. State variable observers were discussed in Chapter 3 of this review. A homing missile guidance problem served as the model for and the application of Vergez' results. In the homing missile problem, the observer was a nonlinear function of the state variables. Secondary emphasis was placed on the design of an improved guidance law based on information developed during the stability analysis.

Vergez found that, for a nonlinear deterministic system involving an optimal controller and a state variable observer, no result similar to the certainty equivalence principle existed. Vergez considered the following questions:

(a) Is it possible to say that the combination of separate Lyapunov functions selected for the controller and the observer provides a valid Lyapunov function for the cascaded system?

(b) If the combination of the separate Lyapunov functions is valid for the closed-loop system, is it the best choice of a Lyapunov function for the composite system in terms of system stability? If not, is there a better Lyapunov function to be selected?

(c) If Lyapunov functions can be found for the cascaded systems of interest, can they be used to analyze the stability of the resulting closed-loop system? Can such functions be used to determine the effects of wide variations in parameter values on the stability of the composite system?

Vergez analyzed a special class of closed-loop control systems composed of a controller and an observer in cascade. The dynamic system was assumed to be linear and time-varying and the system parameters were assumed to be uncertain. The observer was restricted to a design involving a linear combination of the state variables. In this pseudo-linear design the coefficient of each state variable was allowed to be an explicit function of the original measurement. In essence, Vergez proposed a pseudo-measurement algorithm for use as a state estimator. This algorithm takes the nonlinear angle measurement model and transforms that data into a linear estimate of the required system states. A difficult and critical problem in the design of such an observer is the manner in which the target acceleration is modeled. The target acceleration cannot be directly measured. At the same time, the target acceleration, when integrated, directly affects the numerical values of the remaining velocity and position states. For that reason Vergez investigated a target acceleration model with varying parameters[15.4].

The linear-quadratic-Gaussian guidance law has demonstrated the potential for significant missile guidance improvements[15.6]. This guidance law is designed assuming that the missile-to-target position, velocity, acceleration, and time-to-go are all available and are accurately known. None of these factors, except for the missile's own acceleration, are directly available on-board a homing missile. To develop estimates of these unknown values, estimation algorithms based on the Kalman filter have been studied[15.7].

For homing missiles having passive angle-only seekers, these estimation algorithms have not shown themselves to be very successful in terms of accurately estimating the required state information. However, in these same applications the linear quadratic guidance laws have been successful in producing small miss distances. This optimal guidance law could produce even smaller terminal miss distance values if the state variable information were accurately known. One goal of Vergez' research was thus to design a linear-quadratic-Gaussian guidance law which strives to minimize the terminal miss distance and simultaneously improve the performance of the state variable estimation process. This was done by including a term in the performance measure which maximizes the observability Grammian matrix of the estimation algorithm.

The observability Grammian matrix is a measure of the estimation system's performance. The corresponding term in the performance measure is based on a Lyapunov function selected for the linear time-varying problem.

Two Lyapunov stability methods for the controller and the observer, or state estimator, were first detailed and discussed. The two separate Lyapunov functions were then combined and the result was analyzed to determine whether or not the combination represented a valid Lyapunov function for the composite system formed by the dynamic system, the observer, and the controller. This was done for both the continuous-time and the discrete-time case.

A Lyapunov function was then derived for the composite system under the assumption that the control law was a linear function of the estimated state variables. This would normally be the case in any linear-quadratic-Gaussian control problem. The Lyapunov function used was the expected value of the performance measure. This Lyapunov function was validated through the appropriate stability conditions.

A third Lyapunov function was then derived for the composite system allowing for possible parameter variations. As a first step in this process the parameter uncertainties were identified and incorporated in the linear time-varying model of the dynamic system. The same Lyapunov function, the expected value of the performance measure, was derived and validated.

The linear-quadratic-Gaussian control law which simultaneously minimized miss distance and improved the performance of the state variable estimation process was then derived. A pseudo-measurement estimation algorithm was employed.

Several applications of this approach were then studied by numerical analysis and digital computer simulation. Two linear time-invariant examples were presented and the three Lyapunov functions for each example were developed. The first example was a scalar cascaded system and the second a multi-variable cascaded system. Acceptable ranges of parameter uncertainties were determined so that the system stability was maintained. Variations in the control feedback matrix were then studied and the results were compared to those obtainable from an eigenvalue analysis. This permitted the accuracy of the Lyapunov function's predictions regarding system stability to be compared with results obtained by a different method.

Next, the methods outlined were applied to the design of a linear time-varying system and two examples were considered. The first example was a linear-quadratic-Gaussian missile guidance problem in which the control law was time-varying. The control law was developed as a function of the time-to-go before intercept. The methods were applied to analyze the performance of this system when errors were present in the value of time-to-go and in the values contained in the matrix describing the modeled dynamic system. In the second linear time-varying example, a homing missile system having the ability to obtain angle-only measurements was studied. A pseudo-measurement observer was designed to estimate the required state variable values. The Lyapunov functions developed were used to investigate the performance of this system as affected by errors in target acceleration modeling.

A final example considered the use of the linear-quadratic-Gaussian control law to minimize miss distance and simultaneously maximize the observability Grammian matrix of the pseudo-measurement observer. Vergez' text[15.5] contains 141 references related to optimal control, estimation theory, Kalman filtering, stability, mathematical modeling, and classical control system design.

15.3.2 Conclusions About Lyapunov Functions

The derivations of the Lyapunov functions used in this work were based on a composite dynamic system formed by a linear (possibly time-varying) system, a state variable estimator, and a feedback controller. The numerical results presented by Vergez assumed that the measurements were noiseless, thus the state variable estimator functioned as a state variable observer.

The Lyapunov function which was formed by adding the controller Lyapunov function to the observer Lyapunov function was found to not be valid for all controller/observer systems. However, it was found that the controller performance measure could be scaled so that the combined Lyapunov functions were valid without affecting the resulting controller gains. It was also found that the combined Lyapunov function could be used as a tool for improving the relative stability of the composite system by permitting simultaneous selection of the controller and observer system design parameters.

A different Lyapunov function for the composite system was developed to overcome the problem of validity. The result was a Lyapunov function formed by the sum of the separate controller and observer Lyapunov functions and an added term corresponding to the interaction of the system states and the observer errors. This Lyapunov function was found to be valid for all controller/observer systems but was very sensitive to system parameter variations.

To account for the variations in the system parameters another Lyapunov function was derived. This function was able to accurately predict the stability of the composite dynamic system in the presence of parameter variations as compared to an analysis of the system's eigenvalues. This same Lyapunov function also provided a measure of the control system's performance in the linear time-varying finite-time homing missile guidance example.

A control law was designed based on linear-quadratic-Gaussian theory for the homing missile guidance problem. This control law minimized the terminal miss distance and simultaneously resulted in improved performance of the observer. The use of the Lyapunov function method allowed an analytic solution to the closed-loop control system design problem to be obtained.

15.4 Numerical Methods for the Guidance and Control of Air-to-Air Missiles

Shieh[15.8] presented an indirect method for the numerical solution of a guidance and control problem for a general air-to-air missile.

The dynamic system was modeled as a nonlinear two-person zero-sum differential game. Differential games were discussed in Chapter 12 of this review. The dynamic equations of motion which described the trajectories of the two players (the pursuing missile and its evading target aircraft) employed eight kinematic state variables. Both the missile and the target were allowed to perform 3-D maneuvers.

The control inputs of both players were subject to state-dependent constraints. The information structure of the game was modeled as a nonlinear function of the state variables and different noise sources and distributions were considered. The performance measure was a quadratic form to be minimized by the pursuing missile and maximized by the evading target.

The original problem was decomposed into three simpler but solvable problems:

(1) an open-loop, perfect-information differential game,

(2) real-time state variable estimation problem, and

(3) near closed-loop filter updating problem.

A differential dynamic programming method was applied to solve the differential game problem with state-dependent constraints. This algorithm required the integration of fewer first-order differential equations than other methods and did not require a good initial guess of unknown parameters. The selection of a proper control increment was recognized as a drawback of the differential dynamic programming method. To overcome this difficulty a new algorithm, the retrogressive weighted convergency control parameters algorithm, was developed and applied. This new technique used the local and accumulated control errors as weighting factors to automatically adjust the step size.

The real-time nonlinear state variable estimator developed in response to the second subproblem was similar to the extended Kalman filter. The state variable estimator minimized the variance of the error propagation matrix and used the most recent measurement data to compute the filter updates. The major difference between the state variable estimator developed in this work and the extended Kalman filter involved the inclusion of time lags which accounted for the delay between the time when the measurement data was available and the later time when the filter update was computed. This time delay was found to be a significant factor since the target tracking frequency became very high during the terminal stage of the missile-target engagement.

Numerical simulations were performed for long- and short-range air-to-air missiles. The long-range missile was targeted on a bomber/fighter aircraft. The short-range missile was engaged against a highly-maneuverable fighter aircraft. Many optimal pursuit and evasion tactics were discovered during this process. The indirect solution method outlined in this work is a promising approach toward the realization and implementation of differential game methodology for missile guidance problems.

15.4.1 Requirements for an Air-to-Air Missile

The guidance and control of an air-to-air missile is a challenging application of modern control theory. The challenge arises from the very fast and highly nonlinear nature of the missile and target dynamics during an engagement. In nearly all present missile systems, proportional navigation is implemented as a guidance law because it is relatively simple to implement and is recognized to be effective against a slowly maneuvering target. Future trends in aerial combat will require that advanced air-to-air missiles have a high probability of kill, a launch-and-leave capability, and an all-aspect launch capability. Expected targets will no longer be slowly maneuvering aircraft, but highly intelligent, rapidly maneuvering adversaries.

These requirements for an air-to-air missile have led to increased application of techniques and methods drawn from modern control theory in attempts to derive advanced optimal guidance laws and to estimate the necessary information required for their implementation by sophisticated processing of sensor-produced data.

The general configuration of an air-to-air missile is shown in Figure 15-2. The subsystem functional block diagram of this missile is given in Figure 15.3. The seeker module tracks the motion of the target by either an active or passive means and provides raw sensor data to the guidance and control module. This raw data is processed by the guidance and control module to generate estimated values of those state variables required to compute the best control strategy. This control strategy must be designed to account for dynamic, rapidly changing missile-target kinematics. The autopilot converts the control strategy into steering and, if necessary, propulsion commands. The missile system operates in a closed-loop manner until the target is intercepted or lost.

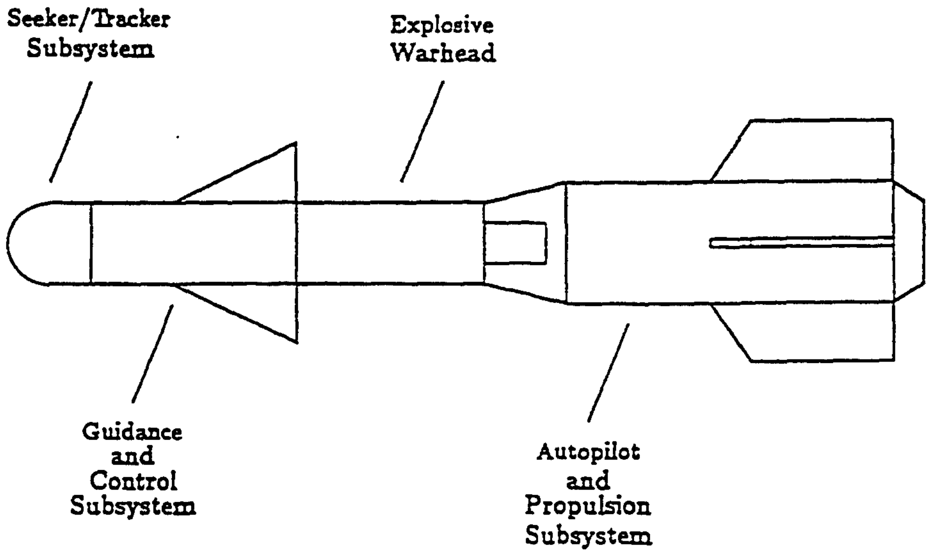


Figure 15.2 General air-to-air missile configuration

It should be noted that a similar structure and block diagram exist for the evader. The evader is aware of the position of the attacking missile based on sensor data or visual observation. The evader uses this information to maximize the distance between the aircraft and the missile. The dynamic equations which describe the motion of the evading target aircraft are, in the context of a differential game, equally as important as those of the pursuing missile.

Sheih's work was restricted to the implementation of a guidance and control subsystem for a general air-to air missile. Other factors such as the design of the seeker and aerodynamic effects were not included. In this work the missile-target engagement scenario was described, general two-person zero-sum differential games were reviewed, the assumptions necessary to arrive at a tractable mathematical model for the problem were stated and the necessary solution techniques were outlined.

The kinematic equations of motion for the pursuing missile and the evading aircraft were developed in detail. The kinematics of two particles moving independently in 3-D space and the target's moving triad and its properties were presented.

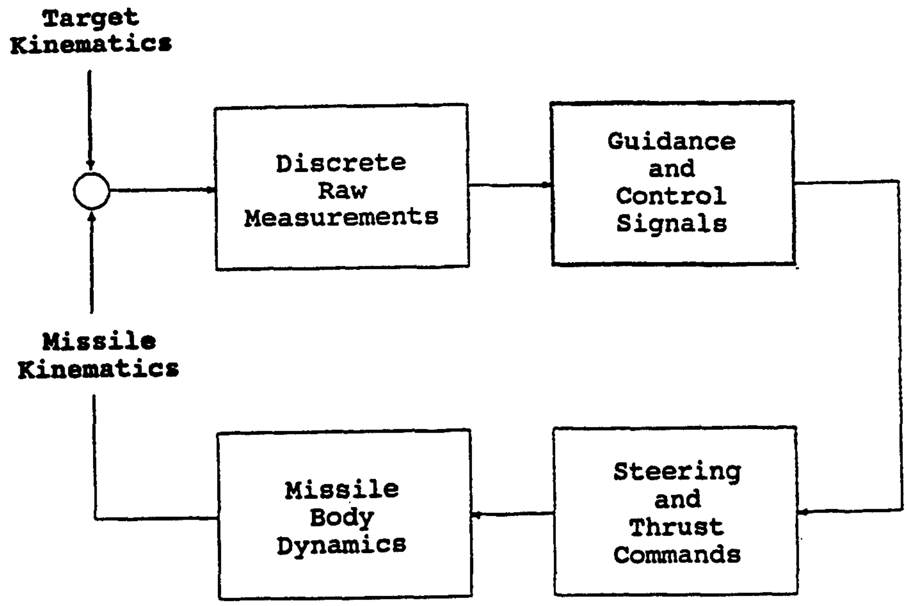


Figure 15.3 Missile subsystem functional block diagram

A deterministic version of the original problem was stated and an extended second-order differential dynamic programming algorithm was derived to solve a transformed version of the deterministic problem. The retrogressive weighted convergency control parameters method was derived and introduced to overcome the problem of finding a proper increment in each iteration of the algorithm.

A new real-time non-linear tracking filter algorithm was developed and implemented as part of the guidance and control subsystem. The imperfect information problem was restated and the extended Kalman filter was reviewed. Shieh's tracking filter algorithm is similar to the extended Kalman filter, but accounts for time lags which exist between the time the raw data is collected by the sensors and the time at which the filter updates are available for control.

Two different missile-target engagements were simulated to demonstrate the effectiveness of the numerical algorithms developed during this effort. The first example was an engagement involving a long-range air-to-air missile encountering a fighter/bomber target. The second example was an engagement between a short-range air-to-air missile and a highly agile fighter aircraft. Three different encounter scenarios were studied in the second example.

15.4.2 An Overview of Two-Person, Zero-Sum, Differential Games

Game theory is concerned with the mathematical modeling of situations in which the participants, or players, have conflicting interests concerning the outcome. If the situation does not evolve over time it may be modeled by a static game; otherwise a dynamic game model is required. A differential game is a model of a situation which evolves over time and in which the dynamic system is described by a set of differential equations. A differential game is an example of a dynamic game.

Research into the application of differential game theory to problems of pursuit and evasion was initiated by R. Issacs at the RAND corporation in 1954. Issac's book on differential games[15.9] was published in 1965 and received widespread attention.

Since the early 1960's many research papers discussing various aspects of the problem of pursuit and evasion between two adversaries have been published. Shieh's thesis contains a bibliography listing over 80 references related to this problem. The analyses in the majority of these papers is quite complicated and the resulting algorithms have been too complex to permit real-time application to the high-speed air-to-air missile problem.

The general structure of a two-person zero-sum differential game can be developed in a highly compact mathematical form. The parsimonious nature of this development provides no indication of the problem's complexity or the difficulty in obtaining a solution even for simple examples. For this reason, it is difficult to cite or develop simple examples of differential games which are directly applicable to the guidance and control of tactical guided weapons.

The zero-sum two-person differential game is played over a time interval beginning at an initial time t_0 (often equal to zero) and ending at some final time t_f (also equal to T). The final time, T, need not be finite. Each of the two opponents, or players, can independently apply a control input to a dynamic system described by the nonlinear differential equation:

$$\frac{dx(t)}{dt} = f\left(x(t),\, u(t),\, v(t),\, w(t),\, t\right),\; x(o) = x_0 .$$

The state variable vector x(t) is n-dimensional, the control input vector of player A, u(t), is m-dimensional, and the control input vector of player B, v(t), is p-dimensional. The vector w(t) denotes the disturbances acting on the dynamic system. These disturbances can arise from a variety of sources including measurement noise. The initial state of the dynamic system is specified by x_0.

The state and control variables are subject to nonlinear inequality constraints of the form:

$$S\left(x(t), u(t), v(t), t\right) \leq 0, \; 0 \leq t \leq T.$$

The control actions applied by each player in this model must be based on each player's own history of observations. The observation history of player A is:

$$y(t) = h\left(x(t), p(t), t-d_1\right)$$

and the observation history of player B is:

$$z(t) = g\left(x(t), q(t), t-d_2\right).$$

In these observation models h(.) and g(.) are nonlinear functions, p(t) and q(t) are observation or measurement noise processes, and d_1 and d_2 are time lags which occur between the time a state sample is measured and the time when the measured data is available for use by the guidance subsystem of player A or B.

The control actions selected by each player are subject to a set of magnitude constraints which take the form:

$$|u(t)| \leq c,$$

$$|v(t)| \leq d.$$

The end of the game occurs at the final time t_f when a terminal condition is satisfied. This terminal condition is mathematically specified by:

$$F\left(x(t_f), t_f\right) = 0.$$

The value of the resulting final time t_f may be specified as t_f equal to T, at which time the terminal condition must be satisfied, or left to be implicitly determined as the first time t when the terminal condition is satisfied.

Solving this two-person zero-sum differential game consists of finding the two control actions u(t) and v(t) such that the performance measure:

$$J(u, v) = E\left[G\left(x(t_f), t_f\right)\right] + \int_{\tau=0}^{\tau=t_f} L\left(x(\tau), u(\tau), v(\tau), \tau\right), d\tau$$

is minimized with respect to player A, the pursuer, and maximized with respect to player B, the evader.

There are many variations to this general structure which lead to special classes of two-person zero-sum differential games. If the differential equations are replaced by a set of difference equations, the integral is replaced by a summation of terms, and the time of control T is broken into a number of discrete time intervals, the game is called a discrete differential game. If all of the system parameters, all of the measurements made by the players, and the running value of the performance measure are known to both players, the game is said to have complete information. If this is not true, the game is said to have incomplete information.

If all the state variables can be observed exactly without noise effects and are instantaneously available to both players, the game is said to have perfect information. A game with imperfect information is a game in which noise effects are present or in which a time delay exists from the time a measurement is taken until the time it is available for use. A deterministic game is a game having complete and perfect information. All other differential games are stochastic games.

Shieh presented a detailed review of solution techniques proposed for various types of pursuit-evasion differential games, and noted that two different mathematical approaches to solving a general differential game have evolved. The first approach is to develop a highly simplified mathematical model of the dynamic system and the game's performance measure. This approach is intended to reduce the problem's mathematical complexity and lead to a closed-form analytic solution for the players' control strategy. If such a solution can be obtained it may lead to further insights regarding the solution of the more complex game originally posed as a problem.

The second approach retains the game's complex dynamic equations, performance measure, and constraints and attempts to apply mathematical optimization methods to develop a numerical solution to the game. Mathematical optimization methods were the subject of Chapter 8 of this report. The difficulty with this approach lies in the complexity of the solution process. Since the form of the solution is not known in advance, complicated numerical methods may be required. Additionally, it is possible that the numerical solution achieved may be a local optimum, rather than the global optimum sought. This is a problem for all mathematical optimization procedures applied to arbitrary mathematical programming problems having unknown solutions. A further disadvantage of a numerical solution is that the result is usually obtained in the form of an open-loop control policy, rather than the closed-loop form desired.

In this work the original problem was decomposed into a set of simpler subproblems which could be solved either analytically or numerically. The solution to the original problem was synthesized by combining the solutions produced for the subproblems. In this way the guidance and control strategies for the pursuing missile and the evading target were generated as the synthetic solution to a nonlinear, two-person, zero-sum, differential game.

Figure 15.4 illustrates the decomposition used. The problem is initially a nonlinear, incomplete, imperfect information, closed-loop, saddle point problem. This problem is first decomposed into a nonlinear deterministic closed-loop saddle point problem and a nonlinear state variable estimation problem. This decomposition is largely heuristic in nature. The separation principle is assumed to apply. The separation principle allows the problem of control and state variable estimation to be treated independently. Intuitively, since neither player has in practice exact knowledge regarding the states of the dynamic system, the best that can be done is to use the information available from a state variable estimation process. The estimated state variables were assumed to be available to both players instantaneously.

The nonlinear deterministic closed-loop saddle point problem was then further decomposed into a nonlinear open-loop saddle point problem and a near-optimal closed-loop updating algorithm. For a fixed set of initial conditions on the state variables, the open-loop saddle point solution must be the same as the closed-loop saddle point solution if both players employ an optimal strategy over the course of the engagement. Intuitively, if either player employs a non-optimal strategy for some period of time, the other player would surely detect the opponent's error and would readjust their own control strategy to take advantage of the error committed. This can be done by comparing the output of the state variable estimator to a reference trajectory generated as a solution to the open-loop saddle point problem. By updating the estimates and reference trajectory sufficiently fast a near-optimal closed-loop control policy can be developed.

An extended second-order differential dynamic programming method was developed and applied to solve the nonlinear open-loop saddle point problem. A solution to the nonlinear state estimation problem was developed in the course of Sheih's work[15.8]. The near-optimal closed-loop updating algorithm was developed separately in a series of papers by Anderson[15.10, 15.11].

15.4.3 Kinematic Equations of Motion

In Sheih's formulation of the missile-target engagement, the dynamic system was represented by a set of kinematic equations describing the motion of a point-mass pursuing missile and a point-mass evading target aircraft, both maneuvering in 3-D space. This kinematic model involves the position, velocity, and acceleration of both players, assumed to be measurable by each player's tracking radar. A more complex dynamic model would include propulsion forces, mass effects, drag and lift, gravitational force, and other parameters which would substantially increase the number of

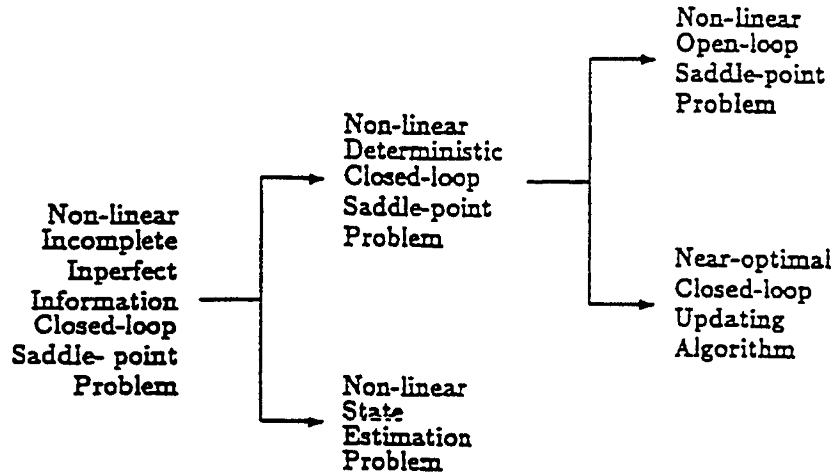


Figure 15.4 Decompositions of the problem

state variables involved in the problem. By focusing attention on the trajectories of motion which result as a solution to the differential game, requirements which must be met by the propulsion systems, airframes, and guidance and control module can later be developed.

Rather than modeling the kinematics in terms of the translational and angular positions of both players, and then developing a set of state variable equations for the 12 resulting degrees of freedom (the translational and angular positions and velocities of both players), a set of three generalized coordinates, the Euler angles, was associated with each player, for a total of 12 state variables, and a pair of generalized coordinates defining the players' speed was introduced, resulting in a total of 14 state variables. By examining the relative motion between the pursuer and the evader, a set of eight state variables can then be obtained. This analytic process substantially reduced the complexity and computational requirements of the solution processes.

15.4.4 Solution of the Deterministic Nonlinear Open-Loop Saddle Point Problem

In a game with complete information, each player has exact instantaneous knowledge of the state of the game, the opponent's goal, and the capabilities and limits of the opponent's maneuverability. Since no random effects exist, a differential game of this type is deterministic and can, in principle be solved analytically.

The deterministic differential game which resulted from the decomposition of the original problem was governed by a set of first-order nonlinear differential equations:

$$\frac{dx(t)}{dt} = f\left(x(t),\, u(t),\, v(t),\, t\right),\; x(0) = x_0.$$

In this dynamic system the state variable vector x(t) is n-dimensional, the pursuer's control vector, u(t), is p-dimensional and the evader's control vector, v(t), is q-dimensional. The starting time, t_0, (equal to zero) and the initial state, x_0, were assumed to be known by both players. The pursuer strives to minimize the performance measure (effective miss distance) and the evader strives to maximize the performance measure. The performance measure employed was:

$$J(u,\, v) = G\left\{x(t_f),\, t_f) + \int_{\tau=0}^{\tau=t_f} L(x(\tau),\, u(\tau),\, v(\tau),\, \tau\right\},\; d\tau.$$

The final time t_f was determined implicitly by the solution to a set of m algebraic terminal conditions:

$$F(x(t_f),\, t_f) = 0.$$

The control inputs were subject to a set of time-varying, state-dependent constraints:

$$C_i\left(x(t),\, u(t),\, t\right) \le 0,\; i = 1,\, 2,\, \ldots,\, p',$$

$$D_j\left(x(t),\, v(t),\, t\right) \le 0,\; j = 1,\, 2,\, \ldots,\, q'.$$

The optimal solution given by the vector functions $u^*(t)$ and $v^*(t)$ must satisfy the saddle-point condition:

$$J(u^*,\, v) \le J(u^*,\, v^*) \le J(u,\, v^*).$$

Anderson[15.12, 15.13, 15.14] applied a first-order differential dynamic programming method combined with Jarmark's Convergence Control Parameter method[15.15] to compute an open-loop saddle-point solution to a somewhat simpler problem. This solution was then used to generate the reference state trajectory for a subsequent update.

In the work presented in Sheih's thesis[15.8] a second-order differential dynamic programming algorithm was developed and applied to the missile-target intercept problem. The details of the differential dynamic programming algorithm and the method by which the retrogressive weighted convergence control parameters technique was implemented are detailed in Sheih's report.

The algorithm consisted of two major steps. In the first step a set of optimal control strategies u^* and v^* were determined for use about a reference state trajectory. This step was called the extremization step. In the second step the solution was adjusted so that the terminal conditions were satisfied. This was called the restoration step. The detailed algorithm was repeated until numerical convergence was achieved.

15.4.5 Solution to the Nonlinear State Estimation Problem

Any guided weapon is subject to a variety of external disturbances, noises, and biases. Since the ultimate performance of a guided missile in terms of miss distance is eventually determined by the target tracking process, the effects of tracking errors resulting from noise and disturbances must be recognized. Thermal noise in the electronic circuitry, biases inherent in the missile instrumentation, mechanical inertia, and servo-system errors all contribute to disturbances internal to the seeker. The ultimate effect of these imperfections can be reduced by a careful design process. Noise, disturbances, and biases can be generated by random environmental effects, low radar signal-to-noise ratios or the effects of radar jamming.

A practical real-time nonlinear filtering algorithm was developed by Sheih to estimate the state variables of the dynamic system from a set of periodically sampled nonlinear measurements:

$$y(k) = h\big(x(k)\big) + z(k),\ k = 1, 2, \ldots$$

where x(k) is the true state at time k, and z(k) is a zero-mean white Gaussian noise process with error covariance matrix E(k). The dynamic system represented by the state x(k) has been defined in terms of a set of first-order differential equations.

The design of this filtering algorithm was based on the extended Kalman filter. The main difference between the extended Kalman filter and the filter developed by Sheih was that Sheih's filter accounts for the delays between the time that an observation of the state has been taken and the later time at which that observation may be used for control purposes. The delay arises because the measurements must be collected, processed, and transmitted to the guidance and control system. These processing time delays were permitted to be different for the pursuer and the observer. The resulting algorithm was suitable for implementation and use in the proposed near-real-time guidance and control subsystem.

Sheih's work included a review of the extended Kalman filter. Sheih's derivation of a real-time, nonlinear state variable estimator was based on the assumption that the separation principle applied to the missile-target intercept problem.

Sheih's filtering algorithm consisted of the following steps:

(1) From time $(t_{k-1} + \delta_2)$ to time $(t_k + \delta_2)$, the estimated state variables evolve according to the following differential equation:

$$\frac{dx'(t)}{dt} = f'(x',u,v,t)$$

$$\approx (x',u,v,t),\ t_{k-1} + \delta_2 \le t \le t_k + \delta_2,\ k = 1, 2, \ldots$$

where the system dynamic equation is:

$$\frac{dx(t)}{dt} = f(x,u,v,t).$$

(2) The error covariance matrix propagates according to the following differential equation:

$$\frac{dQ(t)}{dt} = \left[\frac{\partial f}{\partial x}\right]_{x'} Q(t) + Q(t) \left[\frac{\partial f}{\partial x}\right]^T_{x'} .$$

(3) At time t_k, when the measurement $y(t_k)$ is available, the computation of K_k is initiated:

$$K_k = \left[I + \delta_1 F(t_k) + (\delta_2 - \delta_1) F'(t_k)\right] \cdot$$

$$\left[H(t_k) Q(t_k) H^T(t_k) + \Xi\right]^{-1}$$

where:

∂_1 = a computational delay time,

∂_2 = a computational delay time,

$$F(t_k) = \left[\frac{\partial f}{\partial x}\right]_{x'},$$

$Q(t_k)$ = the error covariance matrix,

$F'(t_k)$ = $F(t_k + \partial_1)$,

$$H(t_k) = \left[\frac{\partial h}{\partial x}\right]_{x'},$$

the observation is given by:

$$y(t_k) = x\left(x(t_k)\right) + \zeta(t_k) ,$$

Ξ = the covariance matrix of the noise process $\zeta(.)$.

15.5 Optimal Guidance of Homing Missiles

Ashida[15.16] studied the application of optimal control theory to the problem of homing missile guidance. The measure of performance in this study was to determine a missile flight trajectory which yielded a minimal miss distance. The guidance problem was outlined, a literature survey was completed, and definitions pertaining to the homing missile guidance problem, features of missile systems and missile control schemes, performance evaluation methods and, the overall problem were presented.

The homing missile guidance problem was originally discussed in terms of the kinematics of the problem. Proportional guidance, a classical missile guidance law, was studied in some detail, and the performance obtained using proportional guidance was compared with that achieved using guidance laws based on optimal control theory.

A model of the homing missile guidance problem based on optimal control theory's linear quadratic theory was described and the analytical solution of linear quadratic optimal control problems was summarized. Ashida developed a modeling technique for the homing missile guidance problem which differs significantly from prior mathematical models and offers certain computational advantages.

A general homing missile guidance problem was stated and the application of optimal control theory was indicated. The problem of realizing the solution of this problem for an arbitrary engagement was addressed. Several engagements in which the initial heading error between the missile and its target was large were studied and analytic solutions were obtained for these problems.

The homing missile guidance problem was then generalized to include target maneuvers, missile speed changes, and a more realistic performance measure. The optimality of proportional guidance systems was investigated for engagements having large initial heading errors.

The homing missile guidance problem can be divided into three subproblems involving the estimation of target and missile state variables, the computation of missile trajectory commands based on a selected guidance law, and control of the aerodynamic missile along its trajectory. State variable estimation in this application involves the determination of the relative target position, velocity and acceleration, and estimation of the time-to-go, the time remaining until the missile impacts the target. The control subproblem requires the design of a suitable autopilot which will stabilize the missile airframe and permit the missile to execute maneuvers commanded by the guidance subsystem. The guidance law uses measured or estimated data regarding the target and missile trajectories to develop commands for the missile autopilot and airframe which will steer the missile along a desired course.

The problem of suitably guiding a homing aerodynamic missile from its launch point to its impact with a target has been studied for over 40 years. During this time period the engagement scenarios have evolved from defending a fixed installation against attack by a long-range slow-moving bomber to high-speed aerial combat and the autonomous attack of highly-mobile surface targets. A variety of guidance laws have been proposed and implemented based on classical control system design methodologies and easily implementable extensions of classical proportional navigation.

Methods drawn from modern control theory have recently provided alternative methodologies for the development and implementation of guidance laws. These modern methods include the application of modern control theory, differential game theory, and numerical methods such as dynamic programming. For many applications classical methods continue to be used.

15.5.1 The State Variable Modeling Problem

Ashida[15.16] proposed and analyzed a new control law for homing missile guidance. The new law was called the proportional bang-bang guidance law. The development of this guidance law was based on the application of modern control theory modeling techniques, and the analytical solution of a pair of optimal control problems. The missile guidance problem was separated from the problems of state variable estimation and missile autopilot design, thus permitting the development of a solution in closed-form useful for analysis and comparison with conventional proportional navigation guidance.

Classically, the design and analysis of a missile guidance system is based on the use of a linearized set of dynamic equations for the airframe linear and rotational motion. A nominally perfect collision course was assumed and the evasive maneuvering of the target was essentially ignored. The application of modern control theory permits more complex engagement scenarios to be investigated. Target maneuvers, missile speed changes, and measures of performance other than miss distance can easily be included in a modern control theory model.

The homing missile guidance problem is characterized by three important factors. First, the angular line-of-sight rate upon which proportional navigation is based grows numerically large during the final few seconds of an engagement unless the missile is precisely on the prescribed constant bearing course. Second, the flight time of the missile is limited by aerodynamics, propulsion and the effect of gravity. The time of control cannot be freely chosen, nor can an infinite time of control always be assumed. The optimal controller for the closed-loop, time-varying linear quadratic problem requires that the time-to-go be determined. Finally, it is necessary to account for target maneuvers, evasive actions, and time-varying missile and target velocities. The general missile guidance problem is thus modeled by a highly nonlinear, time-varying dynamic system, the time of control cannot be determined in advance, and the actions of the opponent cannot always be accurately predicted.

In this work optimal control theory was applied to the guidance of a homing missile. The tracking and guidance problem was confined to a single plane and the problem was treated as a deterministic control problem. Exact knowledge of the missile and target state variables was assumed. Only the dynamics of the autopilot and airframe were included in the analysis. Any dynamics associated with the missile seeker subsystem were ignored. The missile motion was treated as the motion of a point mass, and the missile flight path heading was used as the missile heading angle. The problem to be solved was posed as an optimal control problem with a specified performance measure. The state variable model consisted of a set of angular kinematic equations for the missile and the target and a set of dynamic equations which modeled the missile autopilot dynamics. The optimal control strategy was shown to depend on the manner in which the performance measure was defined and on any constraints imposed on the solution or the state variables. Several specific optimal control problems were considered, including the linear quadratic problem and the time optimal control problem with an initial heading error, this last with a control constraint and with a quadratic performance measure.

Proportional navigation guidance was studied and compared with other guidance laws based on optimal control methods. Proportional navigation guidance is a linear control law in which the commanded missile turning rate is proportional to the measured line-of-sight rate. If the dynamics of the missile or target, a launch error, or a target maneuver is included in the fundamental model for proportional navigation guidance, a non-zero miss distance always results. This does not mean that the missile does not hit the target, but rather that the missile impacts the target at some point other than the desired aim point at the geometric center of the mathematical target model.

The homing missile guidance problem was then formulated as a linear quadratic optimal control problem. For perfect autopilot dynamics, proportional navigation with a navigation gain of three was shown to be identical to the optimal control solution. The solution of the linear quadratic problem exhibits several drawbacks typical of any optimal control problem's solution. The feedback gains which are applied to the system state variables increase indefinitely as the terminal time t_f approaches. For a model including first-order autopilot dynamics, the feedback controller is rather complicated and precise information regarding the time-to-go in this simplified engagement is needed to compute the required control input. When the model was extended to second-order autopilot dynamics, the optimal feedback guidance law becomes overly complicated. When the dynamics of the missile seeker are included, real-time estimation of both the time-to-go and the line-of-sight rate are required. For these reasons substantial differences between the performance of a conventional proportional navigation guidance law and a guidance law based on optimal control theory can be anticipated. Since guidance laws based on a solution to a linear quadratic optimal control problem present major implementation difficulties, a different formulation of the homing missile guidance problem was recommended.

A new formulation of the optimal homing missile guidance problem was presented. This formulation specifically addressed the problem of an initial heading error. The main idea in this new formulation was the achievement of a reference solution, a constant-bearing course, over a finite time interval just prior to the time when the missile impacts its target. By achieving a constant-bearing course, the numerical problems associated with proportional navigation can be overcome. The result is a nonlinear feedback optimal guidance law capable of successful operation in an engagement with a large initial heading error between the missile and its target.

The optimal control problem was broken into two subproblems. The first subproblem is a minimum time control problem with constraints on the control action. The second subproblem is the minimization of a quadratic performance measure based on the square of the control effort. The mathematical statement of the first problem was:

minimize $J_t = \int_{t_0}^{t_1} dt$

subject to:

$$\frac{dR}{dt} = v_T \cos(\beta - \theta_T) - v_M \sin(\beta - \theta)$$

$$\frac{Rd\beta}{dt} = -v_T \sin(\beta - \theta_T) + v_M \sin(\beta - \theta)$$

$$\frac{d\theta}{dt} = Y(p)u$$

$$|u| \le u_o$$

$$A = -Rd\beta/dt = 0 \textit{ for all time } t \textit{ when } t1 \le t \le tf$$

$$R(t_f) = 0$$

The terminal time t_f is implicitly specified as the time of impact. The alignment, A, is zero when the missile has achieved a constant-bearing course prior to impact. The control input, u, is subject to a magnitude constraint, and the autopilot dynamics are represented by Y(p), a polynomial in which the operator p = d/dt represents a time derivative. The target velocity V_T and the missile velocity, V_M, can be time-varying, and the state variables of the kinematics are the range, R, the missile heading angle, T-h, and the target heading angle β. The missile heading angle T-h is controlled by means of the control input u(t) and the autopilot transfer function Y(s).

The second problem to be solved was a parameter optimization problem:

minimize $J_q = \frac{1}{2}\int_{t_0}^{t_1} u^2\,dt$.

This pair of problems was first solved for the case of perfect autopilot dynamics to demonstrate the methodology and provide a comparison with other examples and guidance laws. A small heading error problem version was solved first by means of a set of linearized perturbation equations. The results were then compared with those obtained by means of proportional navigation and generalized proportional navigation. Then the large heading error problem was investigated, again assuming perfect autopilot dynamics.

The optimal controller for the linearized minimum-time problem with the specified initial conditions and control constraint was found to be a bang-bang controller given by:

$$u^* = -u_0\,\mathrm{sgn}\,(h_0) \text{ for } \tau \in [\tau_o,\ \tau_1]$$

$$= 0 \quad \text{for } \tau \in [\tau_1,\ 1]$$

Here τ is a normalized time defined by $\tau = t/t_f$. The dimensionless performance measure is given by $J_t/t_f^* = \tau_1 - \tau_0$, and the terminal time τ_1 was given by:

$$\tau_1 = 1 - (1-\tau_0)^*[1-2^*\ h_0\ (t_f^*u_0)]^{1/2}.$$

At the terminal time τ_1 the heading error equals zero and from that point on a perfect collision course is maintained by applying the control input $u = 0$. In an actual missile guidance system implementation, the control action would be set equal to $\pm u_0$ depending on the sign of h_0, and the control action would be set to zero whenever the line-of-sight rate was sufficiently small. The parameter h_0 depends on the initial values of the state variables, the initial normalized time τ_0, the initial range and heading angle, and the coefficients of the linearized state equations.

As the magnitude of the control constraint u_0 increases, the terminal time decreases as does the time indicated by the performance measure. For any specific missile and flight conditions there will be an upper limit on the magnitude of this constraint because of induced aerodynamic drag. The second subproblem determines an optimal value of the parameter u_0.

The constraint u_0 which was imposed in the first subproblem can be specified so that the aerodynamic losses due to induced drag are minimized:

minimize $J_q = \frac{1}{2}\int_{t_0}^{t} u^2\,dt$, where u is a constant equal to either $\pm u_l$ over the time interval and the time interval is specified as the solution to:

$$\tau_1 = 1 - (1-\tau_0)*[1-2*/h_0/(t_f*u_0)]^{1/2}.$$

The optimal control was found to be:

$$u^* = -2.25 \ | h_0 |) / t_f * \mathrm{sgn}(h_0) \text{ for } \tau \epsilon (\tau_0, \tau f | ,$$

$$= 0 \quad \text{for } \tau \epsilon \quad (\tau_f, 1 |$$

This guidance law was given the name proportional bang-bang guidance. For a given small initial heading error the optimal control u* was based on an analytical solution to a time-optimal quadratic performance measure optimal control problem. The terminal time can be evaluated in advance, and the magnitude constraint u*0 depends only on the tracking error and the time of control, not on the kinematic relationships which determine the missile trajectory. For the particular example presented, this new control law resulted in a final time 11.25% larger than that for proportional guidance, but required 25% less maximum control. The control effort is not constrained in the usual solution of the proportional navigation guidance problem. This meant that the new guidance law permitted a smaller inner launch envelope for a specified maximum missile turning rate. The new control law can also achieve a perfect collision course.

The above effort subdivided the original problem of homing missile guidance into two subproblems for which solutions could be obtained by means of modern control theory. The first subproblem established a time-optimal control problem with a constraint on the magnitude of the applied control effort. The second subproblem was a parameter optimization problem in which the optimal limit on the applied control effort was determined.

The success of this modeling effort depended on the development and application of a somewhat different state variable model proposed for the dynamic system consisting of the target, the missile and the missile autopilot. The state variable model was constructed so that its two parts could be described separately. The first part consisted of a set of kinematic relations defining the motion of a point-mass missile and a point-mass target in a single plane. The missile autopilot, which accepts control input commands generated by the guidance system and attempts to steer the missile along a desired trajectory, was modeled separately.

This state variable model was claimed to be very general and capable of providing substantial insight into the general homing missile guidance problem. The advantages of using this particular state variable model were:

(a) Any engagement (in the plane) can be investigated using the proposed model.

(b) Since the missile autopilot dynamics and the kinematic equations of motion in the plane were described independently, the missile guidance problem and the state variable estimation problems were not coupled by the kinematics.

(c) Velocity equations were developed to model the missile and target kinematics. These velocity equations were differentiable, providing mathematical and practical convenience.

(d) The dynamic equation for the missile autopilot was selected so as to represent a transverse acceleration command. In this model the control variable u did not represent any particular physical action such as a control surface deflection or a thrust vector orientation. The control variable was not required to be a continuous time function, and switching instantaneously from one limit to the other was allowed.

(e) The state variable model proposed in this effort can be used to represent any homing weapon, such as an aerodynamic missile or an underwater torpedo.

15.5.2 The Homing Missile Guidance Problem

Classical proportional navigation guidance and a guidance law based on the linear quadratic optimal control problem were reviewed and compared with the results of the new modeling technique, and several important observations were made:

(a) Proportional navigation guidance with an effective navigation gain of three provides an optimal guidance law for the proposed model in the special case of small initial heading errors. For the case of large heading errors, proportional navigation is nearly optimal. Optimality was measured in terms of miss distance.

(b) Essential differences between proportional navigation guidance and optimal feedback guidance based on the solution of a linear quadratic optimal control problem were noted. Proportional navigation guidance derives a guidance command based on a measurement of the line-of-sight angular rate. Optimal linear feedback guidance derives a guidance command based on a measurement of the system state variables and the application of a set of time-varying gains.

(c) The solution to the homing missile guidance problem becomes complicated when the autopilot dynamics are involved. The solution to a general linear quadratic optimal control problem usually results in a feedback structure with time-varying gains. These gains become numerically very large as the problem's terminal time is approached. To perform a table look-up of these gains, the time-to-go must be estimated.

The solution can only be developed in a manageable form if the problem remains linear.

(d) A solution based strictly on a linear quadratic formulation of the problem does not employ a practical performance measure unless the autopilot is modeled by a simple gain function. A performance measure based on the missile's turning rate was claimed to have more physical significance than a performance measure based solely on the applied control effort.

(e) When noise is introduced into the process of measuring the state variables, a need to estimate the state variable values arises. The complete solution then requires the simultaneous design of an estimator and a controller to achieve maximum performance.

The homing missile guidance problem was restated and a new homing missile guidance law, proportional bang-bang guidance, was developed. The advantages claimed for this guidance law were:

(a) The proportional bang-bang guidance law was simple, general, and presented no implementation difficulties.

(b) The problems that arise due to numerical singularities as the angular line-of-sight rate rapidly increases or the time-to-go decreases were eliminated. No estimate of time-to-go is required to implement this proposed guidance law.

(c) Proportional bang-bang guidance was able to provide a constant bearing course against a maneuvering target and in the presence of missile speed changes by switching to an equalization technique developed during this effort.

(d) Proportional bang-bang guidance resulted in a nonlinear feedback controller, versus the linear feedback controller obtained when using proportional navigation or an optimal linear quadratic formulation. The proposed method was claimed to allow more freedom for the solution of the state variable estimation problem.

(e) For the specific model investigated, proportional bang-bang guidance was found to yield smaller inner launch envelopes than those obtainable by the use of proportional navigation or optimal linear guidance.

Two disadvantages for proportional bang-bang guidance were noted:

(a) The complete problem is very difficult to solve for a model having higher than first-order autopilot dynamics. The reason for this is that the first subproblem to be solved is a linear time-optimal control problem. This problem's solution results in a switching surface which forms a division between several regions in the state variable space. When the number of state variables is low (less than or equal to three) it is often possible to find an analytic expression which explicitly determines the bang-bang nature of the time-optimal control action as a function of the state variables of the system. As more state variables are introduced to account for higher-order autopilot dynamics, the solution of the time-optimal control problem becomes complicated and numerical methods must be employed to determine a solution.

(b) To determine the switching times for the control action and the terminal time of the engagement the proportional bang-bang guidance law requires accurate state variable information. A state variable estimator which provides highly accurate estimates of the state variables is required to obtain a near-exact constant bearing trajectory.

15.5.3 Implementation of Proportional Bang-Bang Guidance

The implementation structure of proportional bang-bang guidance for the homing missile guidance problem considered in this effort is shown in Figure 15-5.

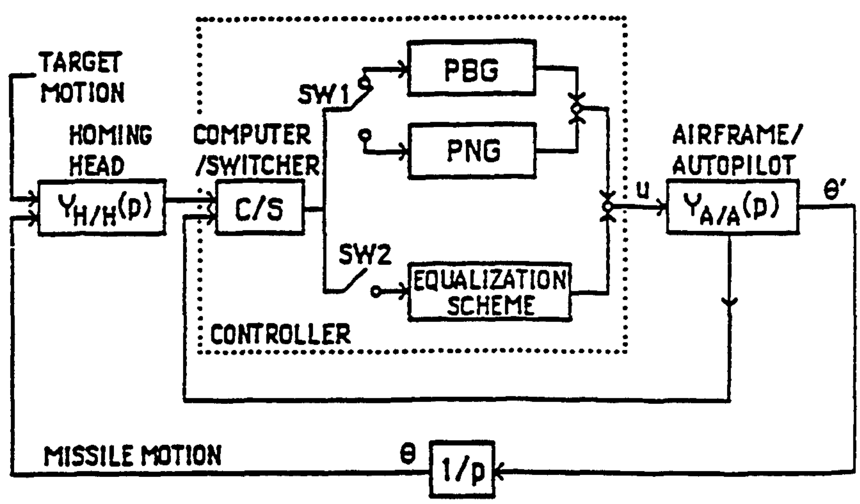


Figure 15.5 Schematic diagram for the implementation of proportional bang-bang guidance (PBG)

During the initial part of the missile's flight the proportional bang-bang guidance law is used to bring the homing missile to a constant bearing course in the minimum possible time. When the constant bearing course is attained the guidance law is switched to classical proportional navigation plus an equalization method. This process guides the missile along the constant bearing course until it impacts its target. The advantages claimed for this implementation were:

(a) The controller is required to compute the terminal time, the control constraint, u_0, and its sign, evaluate the time-optimal switching condition and switch to the proportional navigation plus equalization method. These computations are simple and do not impose a high computational burden.

(b) Precise state estimation is required only in order to compute the terminal time of the engagement and at those times when the switching condition is evaluated. Precise knowledge of the system state at all times is unnecessary.

This effort was devoted to the study and development of a guidance method for use in the plane, a 2-D problem. The results can be applied to a 3-D homing missile engagement since the tracking error is a vector. If a skid-to-turn airframe is involved the tracking error in each plane of control must be evaluated and the missile must be roll stabilized. If a bank-to-turn airframe is involved a roll autopilot is required. This roll autopilot orients the control vector in the direction parallel to the tracking error vector. For either 3-D application, effort must be devoted to the design of an appropriate autopilot and a suitable state estimator.

15.6 A Study of Maximum Information Trajectories for Homing Missile Guidance

Tseng[15.17] investigated the development of a guidance law to enhance the performance of a navigation filter on-board a homing missile. A performance measure representing the information content of the measurements taken along the trajectory was developed and a guidance law which maximized that performance measure along a 3-D trajectory was investigated. This optimal trajectory was called the maximum information trajectory. Realistic missile dynamics were included in the mathematical model. The performance of the navigation filter was found to be better along the maximum information trajectory than along the standard proportional navigation trajectory.

A 2-D maximum information guidance problem served as a basis for the attempted development of a simple guidance law for real-time applications. Constant missile and target velocities were assumed and two optimal control problems, one involving steering angle control and the other normal acceleration control, were solved for the two cases of free final time and fixed final time. The solution of the free final time steering angle control problem was found to yield an infinite information index while the fixed final time steering angle control problem was found to yield a chattering control solution. The solutions obtained for these two problems were not found to be useful for practical application.

A linear quadratic approximation of the maximum information trajectory for the normal acceleration control problem was also investigated. This solution was not found to improve the guidance situation after the missile became unable to observe the target state.

By penalizing the information index with a control effort term, a weighted information problem for the normal acceleration control problem was formulated. Optimal solutions of this problem were obtained for selected weighting valued for both the free final time and the fixed final time cases of this problem. The optimal solutions were examined to reveal characteristics of the maximum information trajectory. The results indicated the worth of further studies of optimal guidance laws based on the simple dynamic models considered in this effort.

This research covered two main subjects: a study of the feasibility of a maximum information guidance concept and an investigation of the development of a practical maximum information guidance law based on a simple 2-D kinematic model.

The maximum information trajectory for a 3-D launch scenario and a dynamic model containing submodels for the missile aerodynamics, propulsion, mass, and an extended Kalman filter was obtained by means of a numerical parametric optimization process. The standard proportional navigation trajectory for this same dynamic system was also computed. Noisy measurements were then generated along both trajectories in order to study the performance of the guidance laws (navigation filter).

To evaluate the performance of the two navigation filters, error histories along both trajectories were computed and compared. The same initial target position, velocity, and acceleration errors were assumed for both trajectories. The error histories were calculated by averaging ten Monte-Carlo runs along each trajectory. It was observed that along the proportional navigation trajectory, the target position and velocity errors diverged and the target acceleration error converged to an incorrect value. Along the maximum information trajectory all of these errors converged to zero.

Further investigation showed that, with more accurate measurements, the filter converged along both trajectories. With less accurate measurements the filter converged along the maximum information trajectory, but along the proportional navigation trajectory the filter converged.

The tracking error along both trajectories diverged when process noise was present, but the tracking error along the maximum information trajectory was less than the tracking error along the proportional navigation trajectory. It was concluded that maximum information guidance enhanced the ability of the filter to track a target compared to proportional navigation guidance. The improvement in filter performance was taken as an indication of the feasibility of maximum information guidance.

It was also observed that in the solution computed for this problem, the missile and target velocities were numerically equal at the time of intercept. This was indicated as an undesirable situation and the need for a means to assure that at intercept the missile velocity remains higher than the target velocity. In this effort the missile-to-target velocity ratio was assumed to be 1.3.

A simple 2-D kinematic model was used as the basis for the development of a practical maximum information guidance law. The missile velocity was assumed constant and either steering angle control or normal acceleration control was permitted. The target was assumed to move in a constant direction at a constant velocity. Both free and fixed final time cases of these control problems were considered. Only the position-related trace information index was retained in the performance measure.

An analysis of these four cases showed that the solution to the free final time steering angle control problem yielded an infinite information index. The trajectory indicated that the missile intercepted the target at an infinite final time. For the fixed final time steering angle control problem the trajectory indicated that the optimal control solution required a chattering control action. Neither of these solutions was obviously useful in a practical implementation.

A control effort term was added to the performance measure for the normal acceleration control problem. When used alone, this control effort term leads to a solution of the minimum control effort problem. The minimum control effort problem had multiple solutions which were verified by testing against the sufficient conditions for optimality. A shooting method was used to solve the weighted information problem and the optimal solution of the minimum control effort problem with larger information content was taken as the initial guess.

An attempt was made to increase the magnitude of the weighting factor so that the information term in the performance measure dominated the control effort term, but the numerical results indicated that the weighting factor could be increased only to a limiting value. Beyond that limiting value no solution to the optimal control problem was found. It was concluded that the weighted information term alone did not have a maximum for either the free or fixed final time normal acceleration control problem.

The feasibility of a maximum information trajectory for a trace information problem with a 3-D dynamic model containing thrust and aerodynamic forces was established. The fixed or free final time trace information problems for a constant velocity missile with steering angle control had no analytical maximums. A performance measure having a weighted information term was introduced for this problem with normal acceleration control, and numerical solutions were found for a range of weighting values. The weighting factor was increased and the optimal weighted information trajectory was examined to determine characteristics of the maximum information trajectory. A realistic mathematical model of the missile was used.

The basic idea behind the study of maximum information trajectories is to develop a guidance law which maximizes the information content along the optimal trajectory. Further study regarding the development of maximum information trajectories based on assumed simple models for the dynamic system was recommended.

15.7 Summary

Optimization of flight trajectories for aerodynamic missiles and conventional aircraft continues to be a challenging topic. Advances in sensors, electronics, and computer technology have led to renewed interest in the on-board generation of real-time guidance commands which control and optimize flight trajectories.

Despite the increased computational speed and memory capacity of current microcomputer systems, fast and efficient numerical algorithms are still required to compute optimal or near-optimal trajectories. Unfortunately the highly nonlinear two-point boundary value equations which yield the solution to optimal atmospheric flight trajectory problems are computationally complex and burdensome. For many problems of interest it is possible to generate accurate open-loop solutions by solving these equations using sophisticated numerical methods. Although these numerical methods are of great value and enable many interesting problems to be solved off-line, their intricacy and computational expense make them virtually useless for real-time applications. Furthermore, onboard real-time guidance requires that the optimal control actions be expressed in closed-loop feedback form as functions of the dynamic system's state variables. It is not necessary that the feedback mechanism be linear, only that it be tractable and computable within an allowable sample time. Since the availability of such solutions for most problems of interest is highly unlikely, an approximate closed-form solution to a less complicated problem is sought for and, if found, used as the basis for developing a sub-optimal or nearly-optimal control policy.

Closed-loop control policies can only be obtained for dynamic systems modeled by highly simplified systems of differential or difference equations. Reduced-order models, in which fast system dynamics having small effects on the dynamic system's behavior are ignored, have thus received considerable attention along with methods for approximating higher order dynamic systems by models of reduced order.

Different approaches were addressed: disturbance accomodating controllers; waveform mode descriptions; closed-loop system analysis using Lyapunov stability; numerical methods optimal guidance using proportional bang-bang guidance; and maximum information trajectories. One of the major lessons learned from this review is that a number of excellent Ph.D. dissertation studies have been performed that apply modern control theory.

REFERENCES

15.1 Etkin, B., Dynamics of Atmospheric Flight, John Wiley & Sons, Inc., New York, 1972.

15.2 Etkin B., Dynamics of Flight - Stability and Control, Second Edition, John Wiley & Sons, Inc., New York, 1982.

15.3 Okata, K., Modern Control Engineering, Prentice-Hall, Inc. Englewood Cliffs, New Jersey, p.207 (1970).

15.4 Geering, H. and Baser, T. Existence of Dominant Solutions in Linear Output Feedback, Proceedings of the 23rd Conference on Decision and Control, Las Vegas, Nevada, December 1984.

15.5 Vergez, P.L., Closed Loop System Analysis Using Lyapunov Stability Theory, Ph.D. Thesis, The University of Texas at Austin, 1987.

15.6 Riggs, Jr., T. and Vergez, P.L., Advanced Air-to-Air Missile Guidance Using Optimal Control and Estimation, AFATL-TR-81-56, June 1981.

15.7 Hull, D., Speyer, J., Tseng, C., and Larson, S., Maximum Information Trajectories for Homing Missile Guidance, AIAA Journal of Guidance, Control and Dynamics, Vol. 8, No. 4, July-August 1984.

15.8 Shieh, F., Numerical Methods for the Guidance and Control of Air-to-Air Missiles, Ph.D. Thesis, University of California at Los Angeles, 1987.

15.9 Issacs, R., Differential Games, John Wiley & Sons, New York, 1965.

15.10 Anderson, G.M., A Status Report on the Development of Near Optimal Differential Game Feedback Methods for Realistic Nonlinear Problems, Proceedings of the IEEE Conference on Decision and Control, 1977.

15.11 Anderson, G.M., Optimal Terminal Aircraft Escape Maneuvers Against Intercepting Missiles, Proceedings of the IEEE Joint Automatic Control Conference, 1981.

15.12 Anderson, G.M., Application of Near-Optimal Differential Game Feedback Strategies to Aircraft Weapon Delivery Maneuevers, IEEE NAECON, 1977.

15.13 Anderson, G.M., Feedback Control for a Pursuing Spacecraft Using Differential Dynamic Programming, AIAA Journal, 1977.

15.14 Anderson, G.M., Optimal Terminal Escape Maneuvers Against Intercept Missiles, IEEE Joint Automatic Control Conference, 1981.

15.15 Jarmark, B.A.S., Differential Dynamical Programming Techniques, Journal of Control and Dynamic Systems, Vol. 17, 1981.

15.16 Ashida, S., Optimal Guidance of Homing Missiles, Ph.D. Thesis, The University of Michigan, 1987.

15.17 Tseng, C.Y., A Study of Maximum Information Trajectories for Homing Missile Guidance, Ph.D. Thesis, University of Texas, December 1983.

This page is intentionally blank.

CHAPTER 16
GUN FIRE CONTROL

16.1 Definitions

The applications of modern control theory to precision guided munitions include both missiles and guided projectiles. Many of the aspects of control theory for missiles included performance requirements of the launch platform. The purpose here is to discuss the launch platforms used for projectiles, whether the projectiles are guided or not. This chapter was condensed from a GACIAC state of the art review edited by Harold H. Burke[16.1].

Many of the basic concepts of modern control theory discussed earlier in this review are applicable to gun fire control systems. The function of a gun fire control system is to offset the gun line from the target line-of-sight, causing a projectile to intercept the target a time-of-flight after firing the gun. In other words, the end result of the gun fire control system's solution is to have a projectile that has been previously fired impact the target that was sighted on a time of flight earlier, as indicated in Figure 16.1. Two classes of target conditions are possible; stationary and moving. Moving targets can be further divided into maneuvering and non-maneuvering targets.

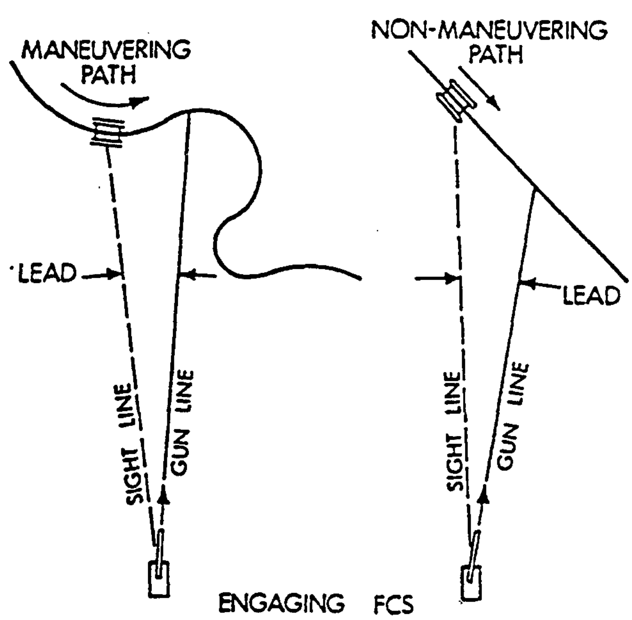


Figure 16.1 Fire Control System and Target Movement.

Non-maneuvering targets are characterized by constant speed and heading motion and maneuvering targets are characterized by non-constant speed and/or heading motion. The distance between the target and the engaging fire control system determines the projectile time of flight. Target motion can be at any arbitrary aspect angle with respect to the LOST between the target and fire control system. The fire control solution must be obtained over a short time interval for maneuvering targets. For non-maneuvering targets, this time interval can be significantly longer than when maneuvering targets are engaged.

Development of a fire control solution for the projectile to hit the target depends upon the prediction and estimation of a variety of conditions that are subject to errors. Figure 16.2 illustrates some of these errors. The error in the ability of a fire control system to cause a projectile to intercept the tracked target a time of flight later is referred to as total gun pointing (TGP) error. The TGP error is made up of the errors occurring in the system, or system induced (SI) errors and target induced (TI) errors, induced by target motion during the time of flight of the projectile to the target. A detailed definition of these errors and the parameters they relate to is given below:

TGP ERROR: Error in gun pointing is defined as the difference between the actual gun pointing direction (at round exit) and the orientation of the target centroid at time of round impact. The ideal gun orientation is the direction which the fire control system must launch a projectile to hit the target centroid.

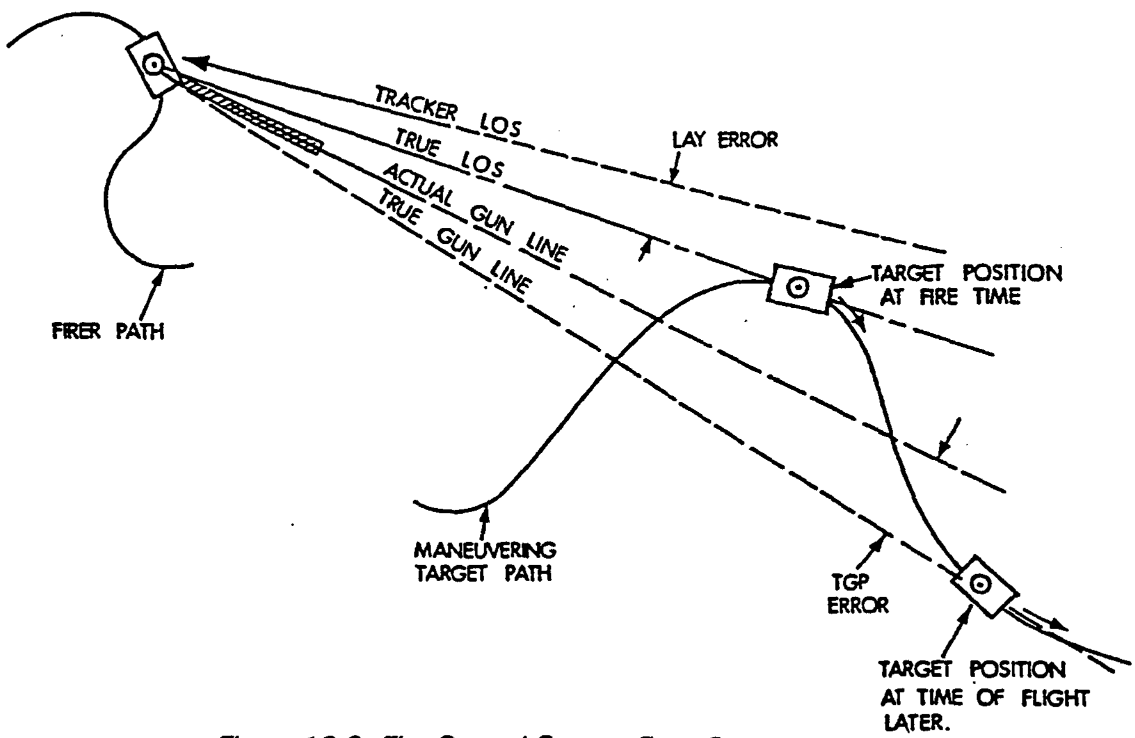


Figure 16.2 Fire Control System Error Sources.

SI ERROR: Error in gun pointing caused by sight pointing not being in coincidence with the true LOS to the target at the time of firing plus the difference between estimates of the LOS movement and the true LOS movement, at the time of firing, propagated through a projectile time of flight (t_f). Other system-induced errors may be caused by the condition of the gun tube and mount.

TI ERROR: Error in gun pointing caused by the target maneuvering during projectile flight time. It is dependent on the order of prediction process of the fire control system. For the first order lead system, the TI Error is the difference between the actual LOS movement during a projectile-time-of-flight and the propagated LOS movement, assuming perfect LOS rate, at the time of firing. For a second order lead system, the TI Error is the difference between the actual LOS movement during a projectile-time-of-flight and the propagated LOS movement, assuming perfect LOS rate and perfect LOS acceleration, at the time of firing. For a predictor order higher than second order, knowledge of a change in LOS acceleration during the flight time of the projectile has the potential to reduce the TI error.

These errors are related in the following manner:

TGP ERROR = SI ERROR + TI ERROR (1)

For a First Order Predictor System

SI ERROR = f(track error, track rate error, t_f) (2)

TI ERROR = f (acceleration of target at firing, acceleration change during projectile t_f) (3)

For a Second Order Predictor System

SI ERROR = f (track error, track rate error, track acceleration error, t_f) (4)

TI ERROR = f (acceleration change during projectile t_f) (5)

and for a Higher Order Predictor System

SI ERROR = f (track error, track rate error, track acceleration error, t_f)

TI ERROR = f (unaccounted for acceleration change during projectile t_f)

The first order predictor system ignores both the presence of target acceleration at time of firing and acceleration change during projectile flight time while the second order predictor system accounts for target acceleration at time of fire but ignores the target acceleration change during projectile flight time. The higher order predictor system attempts to account for acceleration changes during projectile time of flight.

Target-induced errors are functions of target maneuver characteristics, projectile time-of-flight and prediction order. For a given prediction order and with perfect knowledge of the target's state and time-of-flight, the resulting target induced errors represent lower bound TI errors. These prediction errors may become smaller for decreased times-of-flight and with the use of higher order prediction. The use of higher order prediction imposes burdens on the estimation of target motion derivatives with improved performance being directly related to low measurement noise.

16.2 Gun Fire Control Processes

The functioning of a fire control system may be broken down into four distinct processes that are indicated in Figure 16.3. Each of these processes is present in all types of fire control systems. They are: tracking, estimation, prediction, and gun pointing. In specific designs these four processes are accomplished in different manners.

The tracking process is important in all four cases. For the moving firer cases, tracking becomes more critical because the base motion of the firer must be compensated and it may be affected in a secondary manner by target motion. Tracking is usually accomplished manually and is concerned with the alignment of the sight reticle with the target. The gunner is involved directly at this stage and accuracy of tracking will be a characterization of the ability of any given gunner to perform the task. Test data obtained from experimental investigations can be used to determine tracking error means, standard deviations, and correlation time constants useful for building models of the tracking errors.

The estimation process is the intermediate stage between the tracking process and the prediction process and its configuration is dependent upon the order of the prediction process. Estimation is the process of filtering the tracking data to provide the necessary target motion information required in the prediction process. The accuracy of the tracking data will influence the performance of the estimation process. The system error induced by the estimation process decreases with improvement in tracking accuracy.

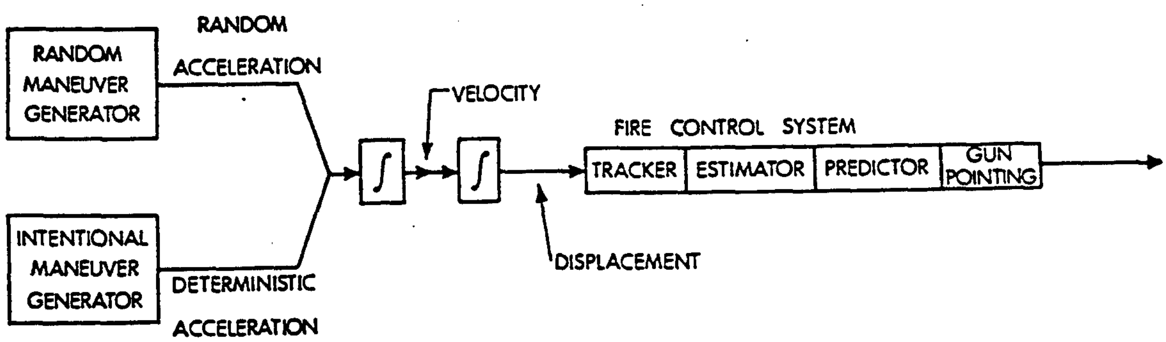


Figure 16.3 Fire Control System Processes.

Prediction of target future position to obtain intercept between projectile and target is dependent upon an estimate of the present motion of the target and the time of flight of the projectile. The output of the estimator is not a complete description of the present motion of the target, therefore, the predictor does not have the necessary information to calculate the target's future position exactly. If restrictions are placed on the allowable threat motions, then the predictor's ability to determine its future position is improved. Oversimplification of allowable threat motions has placed unrealistically simplified requirements on the operation of the estimation and prediction processes. Realistic threat motions are determined by the mobility capabilities of tactical vehicles. In the past, the majority of targets that have been studied have been non-accelerating, i.e., constant speed and heading. The requirements of an estimator and a predictor for this type of motion are to combine the apparent target velocity estimate and projectile time of flight for the lead solution. The required lead is constant and can be realized after some settling time. The existence of accelerating targets requires the system to develop constantly changing lead angles, hence, the need for non-linear prediction.

An important point to observe is that, for the stationary firer-moving target case, the prediction process is required to provide gun command orders that orient the gun to account for target motion during the projectile's time of flight, whereas in the moving firer-stationary target case this prediction process is not required because the LOS existing between the firing point and the target at instant of firing does not move during the projectile's time of flight. For the moving firer-moving target, the LOS also moves after projectile firing.

The gun pointing process is required to align and stabilize the gun along the predicted LOS to the target. The stabilization and the response of the gun pointing loop are major concerns for fire control system performance against maneuvering targets. Stabilization of the gun pointing process could have an adverse effect on overall system performance. The moving firer cases will stress the gun pointing process most severely but it is possible that the gun pointing process will be equally stressed for the stationary firer-moving target case with non-linear prediction.

16.3 Gun Fire Control Configurations

The three basic types of fire control configurations in existence are manual, disturbed reticle and stabilized sight-director systems. They are identified in terms of how each of the fire control processes are mechanized. All existing operational systems utilize the human operator to monitor the difference between the observed target and the reticle and null the error. The degree of participation of the human in each of the types of fire control systems is considerably different. Concern about the stability of the closed loop man-machine system is an important consideration in determining performance and is one of the primary distinguishing features that separates the potential effectiveness of the three types of fire control systems. Figure 16.4 shows the manual fire control system. In the manual system three processes are performed by the man, and the machine serves only to orient the gun line in accordance with the information provided by man.

The disturbed reticle fire control system is shown in Figure 16.5. The four major fire control processes are identified in terms of where in the system each is accomplished.

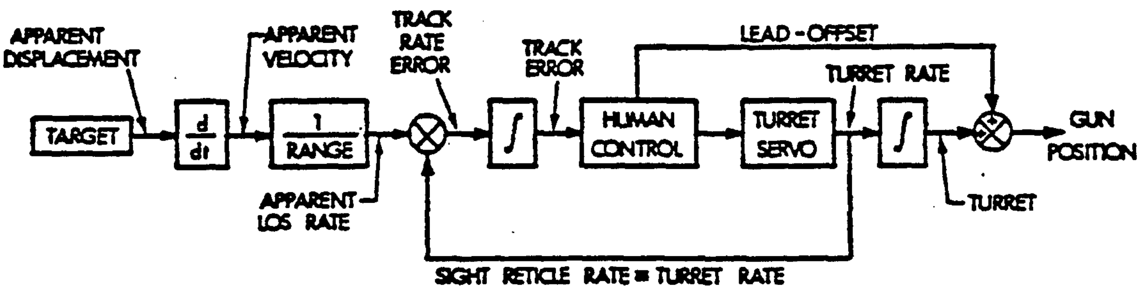


Figure 16.4 Manual Track and Lead System.

Input to the disturbed reticle fire control system is the LOS of the target, θ_T. The human operator moves the handle bar controller to align the reticle of the tracking system with the target. The ability of any human controller to accomplish this task defines the quality of the tracking process. Handle bar controller output, which is directly related to the LOS rate, is used to drive two interdependent subsystems. The first is the turret servo which is commanded to rotate at a rate directly proportional to the handle bar controller deflection. The second subsystem driven by the handle bar controller is a lead screw servo and reticle system. The displacement of the lead screw servo is directly proportional to the filtered handle bar controller deflection multiplied by the projectile time of flight. The output of the lead screw servo is used to offset the reticle of the tracking system from the gun orientation.

There are two distinct feedback signal paths in the disturbed reticle configuration and the human is a series subsystem in both paths. Another important observation is that the signal loop made by the turret servo-man-handle bar controller is a degenerative feedback loop because of the negative summing junction. The signal loop made by the filter, time of flight, lead servo, reticle servo, man, and handle bar controller is a regenerative feedback loop because of two negative summing junctions.

A stabilized sight-director fire control system, shown in Figure 16.6, is actually two distinct systems that are brought together to accomplish the tracking, estimation and prediction processes of a fire control system. Stabilization of the tracking system is independent from stabilization of the turret. The stabilized sight is decoupled from turret and hull motion by the reverse torquing of the outer gimbal of the tracker to account for disturbances of the tracker base which is mounted on the turret. This decoupling enhances the ability of the tracker to maintain coincidence between the sight reticle and the target LOS. The stabilized reticle position can utilize both position and rate feedback to augment the stability of the sight. The orientation of the sight reticle is, therefore, an independent process from the turret motion.

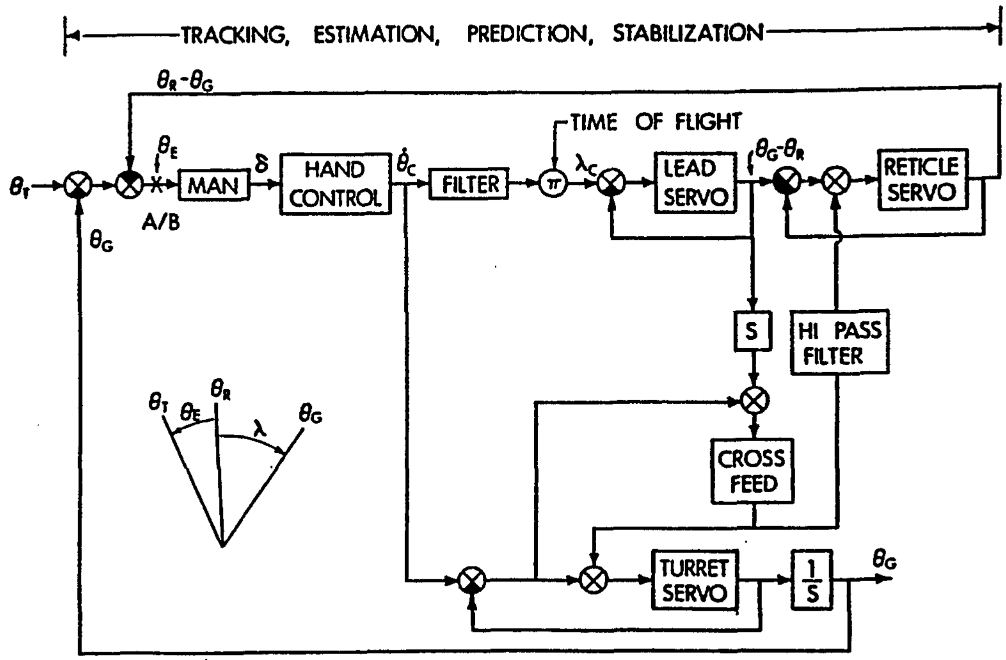


Figure 16.5 Disturbed Reticle Fire Control System.

Position and rate of the LOS are fed to a filter or estimation process to determine the necessary information about the LOS to the target that will be needed to offset the turret servo from the stabilized tracker. Multi-variable, sub-optimal technology can be applied to further improve the quality of tracking that can be realized from the stabilized sight-tracker. Therefore, either linear or non-linear prediction are possibilities for the fire control solutions. If LOS accelerations are to be estimated, the appropriate modeling of target dynamics and tracker uncertainties will be required to insure that the degree of sub-optimality is not excessive. One very significant plus for coupling the estimation and tracking process in a favorable manner is the utilization of sight line rate aiding feedback to the tracker obtained from estimation of the target rates and acceleration. This concept relaxes the task of the human tracker or auto-tracker and will improve the minimization of tracking error.

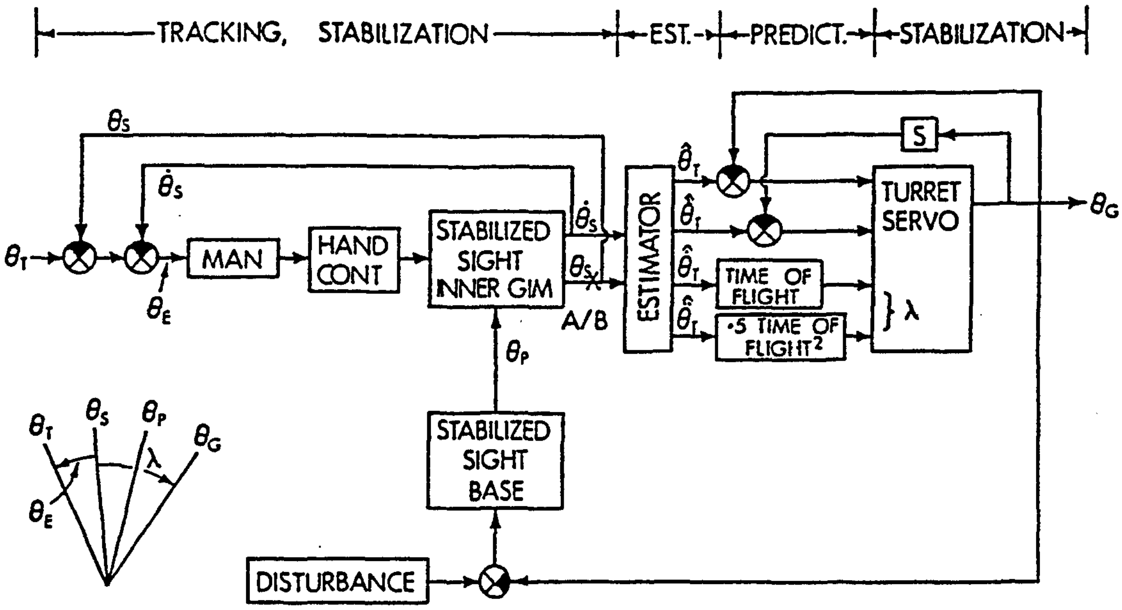


Figure 16.6 Stabilized Sight-Director Fire Control Systems.

Output of the target state estimator is used in two separate paths. The first path uses θ_T and $\dot{\theta}_T$ to drive the turret servo as a director to follow the tracker LOS. The second signal path combines target state estimates with projectile time of flight and offsets the gun from the tracker LOS by the appropriate value to permit intercept of projectile and target a time of flight later.

Performance of the stabilized sightdirector system should not be compromised by maneuvering targets to the extent that the disturbed reticle system is compromised. The basic reason for this is that the tracking system is essentially decoupled from the lead prediction system. However, there are some inherent stabilization problems that can occur in this configuration and they are accentuated by the temptation to obtain high performance of the gun pointing process. The argument goes on as follows: with increased tracker performance, the gun stabilization servo can be made to perform more rapidly, thereby increasing the overall capability of the system. However, with increased performance being required of the turret servo to follow the turret command, the stability of the turret servo may be compromised because of the high gains in the director-follower loop. Experience with similar types of systems has shown that because of non-rigid gun tube and hull structures, the follower loop system must be phase stabilized and not gain stabilized, as is the case for less responsive systems such as disturbed reticle systems. This requires sophisticated compensation circuits to overcome system instabilities.

16.4 Modern Control Theory Concepts

One of the fundamental processes which arises in gun fire control is the process of estimating the state of the target. This estimation process is readily discernable in even the least sophisticated systems. As the system design is augmented to include capabilities against maneuvering targets, the burden upon the estimation process becomes progressively greater both in terms of accuracy and number of states to be estimated. For instance, for straight-line, constant velocity targets, there is no need to estimate acceleration. On the other hand, the utility of a velocity estimate will depend directly upon the accuracy of the estimate. An inaccurate lead may be worse than no lead at all. The same argument holds for the higher derivatives of motion.The conventional approach to the design of estimators and predictors for fire control systems is best illustrated by the following development of models and parameters. One would start by formulating target and observer models of the form.

(a) Target Model

$$X_{k+1} = \Phi_k X_k + B_k U_k$$

(b) Observer Model

$$Y_k = H_k X_k + V_k$$

These models immediately involve a linearization approximation. The target model captures the well defined motion in the state transition matrix, ϕ_k, and leaves the less defined part of the motion to a noise term, $B_k U_k$. The observer is usually a statement that not all the state components are visible, and that the observations are corrupted by error, V_k. (The index, k, is a discrete time index).

If one can further approximate U_k and V_k by white gaussian, zero mean processes, an estimator of X_k can be formulated as:

$$\hat{X}_{k+1} = \Phi_k \tilde{X}_k \qquad \text{(Predicted State)}$$

$$\hat{X}_k = \tilde{X}_k + K_k \; (Y_k - H_K \tilde{X}_k) \qquad \text{(Corrected State)}$$

which is the Kalman Filter wherein

$$K_k = \tilde{P}_k H_k' \,(R_k + H_k \tilde{P} H_k')^{-1} \qquad \text{(Filter Gain)}$$

$$\tilde{P}_{k+1} = \Phi_k \hat{P}_k \Phi_k' + B_k Q_k B_k' \qquad \text{(Predicted Variance)}$$

$$\hat{P}_k = \tilde{P}_k - K_k H_k \tilde{P}_k \qquad \text{(Corrected Variance)}$$

$$R_k = E\,(U_k U_k') \qquad \text{(Observer Noise Variance)}$$

$$Q_k = E\,(V_k V_k') \qquad \text{(Model Noise Variance)}$$

In the most sophisticated fire control systems, the target noise is represented in a target oriented coordinate system. Thus, the given Q_k will rotate as the target moves which in turn leads to a nonsteady K_k. The Kalman gains tend to change throughout the estimation process. In addition, R_k may be range dependent which leads to further variability in K_k. In designing such a filter, the implementor is left with choices of the magnitude of Q_k and R_k (i.e., $\|Q_k\|$ and $\|R_k\|$). A conventional design process would require assessing $\|R_k\|$ from the accuracy of the instrumentation used by the observer. Since $\|Q_k\|$ represents unmodeled behavior, it is usually adjusted to achieve some other objective, such as white innovation, or minimum ensemble miss distances. Whatever the objective, the last phase is unguided by the theory and thus usually requires extensive simulation. The design process for a filter-predictor in tandem is illustrated in Figure 16.7.

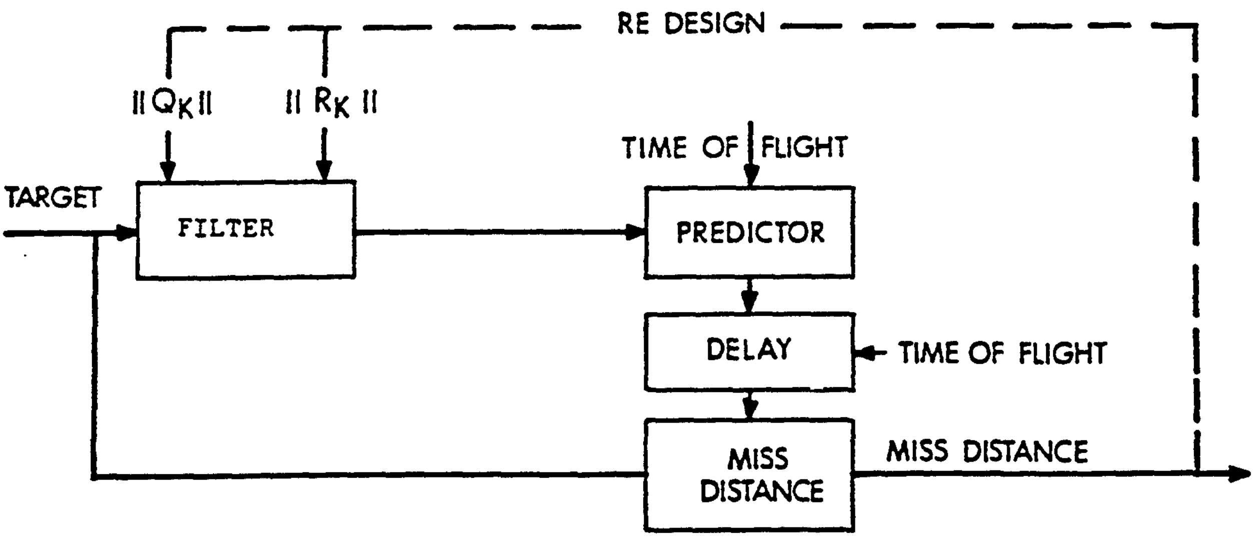


Figure 16.7 Conventional Design of Estimators and Predictors.

The purpose of the predictor in a gun's fire control system is to estimate the future position of the target, necessary information for the development of proper lead angles. Conventional linear prediction, described by the equation

$$\vec{P}_f = \vec{P}_p + \vec{V}_p\, t_F$$

where $\vec{p}_f$ is the future position of the target, $\vec{p}_p$ its present position, $\vec{v}_f$ its current velocity, and t_F the projectile's time of flight, clearly assumes that the target will be flying along a straight line at least for the duration of the projectile's flight. Similarly, conventional parabolic prediction, described by the equation

$$\vec{p}_f = \vec{p}_p + \vec{v}_p \, t_F + (1/2) \, \vec{a}_p \, t_F^2$$

where $\vec{a}_p$ is the target's present acceleration, assumes that the target will be flying along a parabolic arc in the future.

The ability of modern fixed wing aircraft and helicopters to perform evasive maneuvers while carrying out their missions has significantly degraded the performance capabilities of gun systems engaging these targets. These evasive tactics have motivated the initiation of programs to improve the effectiveness of conventional gun systems. One way to improve the effectiveness of a gun system engaging maneuvering targets is to increase its delivery accuracy. A new and different fire control solution using a Circular Arc Aimed Munition (CAAM) prediction concept has the capability of doing just that.

The CAAM predictor makes the assumption that for at least one projectile time of flight (several seconds), an aircraft will fly in a circular arc of fixed radius. This seems to be a reasonable assumption since the laws of aerodynamics and the requirement to maintain a stable operating condition constrain the acceleration vector of the aircraft to remain more or less perpendicular to its velocity vector. In fact, modern fire/flight control systems constrain aircraft to maneuver in sustained (high acceleration) circular arcs during the ordnance delivery portion of the flight profile.

Compared to straight line ordnance delivery, this tactic clearly increases survivability against engagement from an air defense gun equipped with a conventional linear or parabolic predictor. However, it is equally clear that this tactic conforms to the assumption underlying the CAAM concept, and thus should not offer quite as much improvement in survivability against a gun outfitted with a CAAM predictor. Other studies have shown that the CAAM concept will also give improved accuracy against ground targets.

The CAAM prediction concept is given by the equation

$$\vec{p}_f = \vec{p}_p + \vec{v}_p \, T_v \, t_F + (1/2) \, \vec{a}_p \, T_a \, t_F^2$$

where the factors T_v and T_a, which account for the rotational motion of the aircraft maneuvering in a circular arc, are defined by the expressions

$$T_v = 1 - (\Delta\theta)^2 / 6$$

and

$$T_a = 1 - (\Delta\theta)^2 / 12$$

The term $\Delta\theta$ is equal to $| \vec{a}_{p_N} | \, t_F \, / \, | \vec{v}_p |$, the rotation rate of the aircraft (i.e., the magnitude of that component of the acceleration vector perpendicular to the velocity vector divided by the magnitude of the velocity vector) multiplied by the time of flight. This is just the amount of circular arc that the aircraft moves through during the time of flight of the projectile.

All 3D maneuvering trajectories may be divided into segments that are 2D planar trajectories over some time interval as indicated in Figure 16.8. The orientation of the target roll angle about the target velocity in the target body centered intrinsic frame is the cue required to determine the existence and time duration of these 2D segments. Each of these 2D segments of the trajectory becomes a spatially fixed maneuver plane in which the curvature characteristics of the maneuvering target can be assessed.

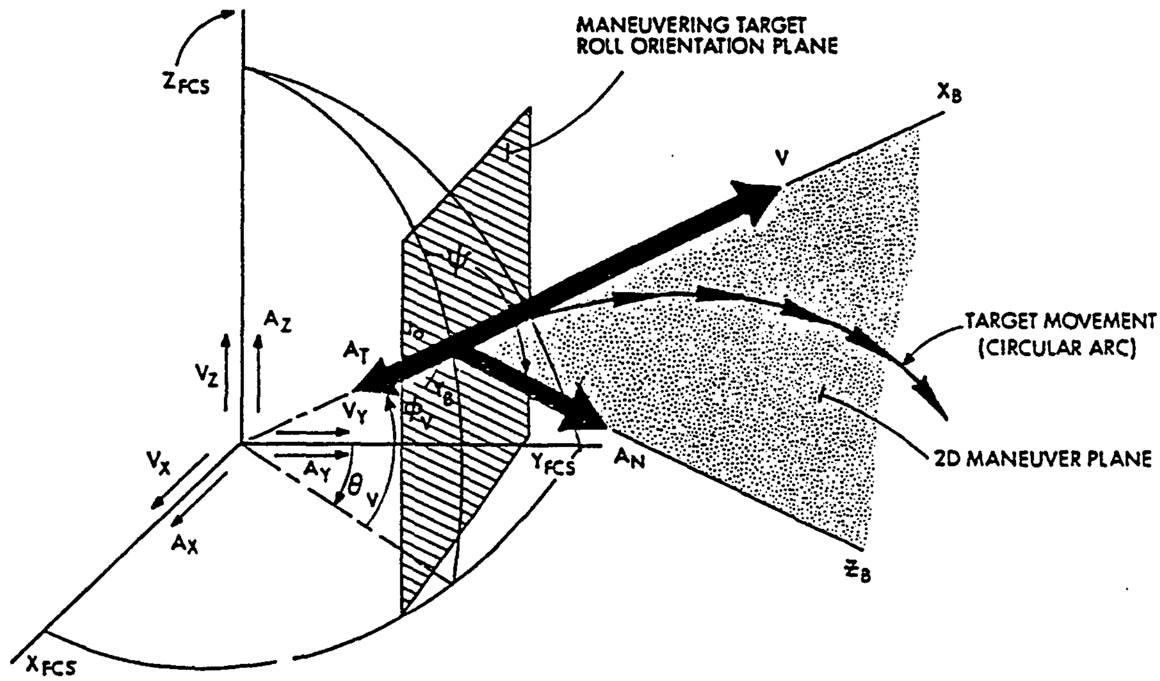


Figure 16.8 3D Maneuvering Target in Fixed 2D Maneuver Plane.

In each of the spatially fixed 2D maneuver planes the curvature of target motion may be approximated with circular arcs whose radi are adjusted with respect to time to account for the variable curvature within each 2D maneuver plane that occurs during a projectile time-of-flight. For a class of maneuvering targets being engaged by a gun FCS using projectiles whose time-of-flights are less than the time duration of the particular 2D spatially fixed planar maneuver, the circular arc approximation is not too restrictive an assumption. The operational time constraints associated with the jinking capabilities of maneuvering targets, for a large class of scenarios exhibiting 3D maneuvering characteristics, are constrained by the mission assignments that permit the target on-board ordnance delivery equipments to satisfactorily perform their functions.

If within each 2D maneuver plane, the maneuvering target is in steady state maneuvering flight, such that the resultant forces acting on the maneuvering target are perpendicular to the vehicle's velocity in the maneuver plane, a flight condition exists which is referred to as a coordinated turn maneuver. The path made by the vehicle when such a balance of forces exists is a circle, and the plane the circle lies in is referred to as the maneuver plane. The velocity of the vehicle is tangent to the maneuver plane circle, the acceleration of the vehicle is normal to the velocity and directed toward the center of the circle. The determination of the movement of the maneuvering target during the projectile time-of-flight, assuming the time-of-flight of the projectile is within the time duration of the maneuver, is then made by the nonlinear CAAM predictor by involving the rationale being presented.

Transformation of target motion, during the projectile time of flight, from the target centered 2D reference frame to the FCS fixed reference frame and then to the LOS reference frame is required in order to accomplish the non linear fire control system goal which is to provide a gun line lead that will improve first round kill probability and maximize kills per stowed ammunition load.

Instead of using a line of sight reference frame, a fire control system (FCS) centered reference frame may be used that does not rely on the 2D maneuver planes in Figure 16.8. Instead, components of target movement during a time flight (tf) are determined with respect to the FCS centered fixed reference frame. Rotational rate of the target (ω) is calculated directly from estimates provided by a nine state Kalman filter, as shown in Figure 16.9. This embodiment of the CAAM concept may be completely realized by making software modifications to existing conventional 2nd order prediction algorithms. This formulation of CAAM prediction shows that the predicted solution will adaptively adjust toward 1st order-linear prediction as ω approaches zero.

16.5 Summary

The critical issue of gun fire control is concerned with the formulation of a high fidelity mathematical model for target state estimation and prediction of target future position a time of flight later. Since the introduction of modern estimation methodology, which is centered around the application of Kalman filtering theory, the target state estimation part of this two-part challenge has

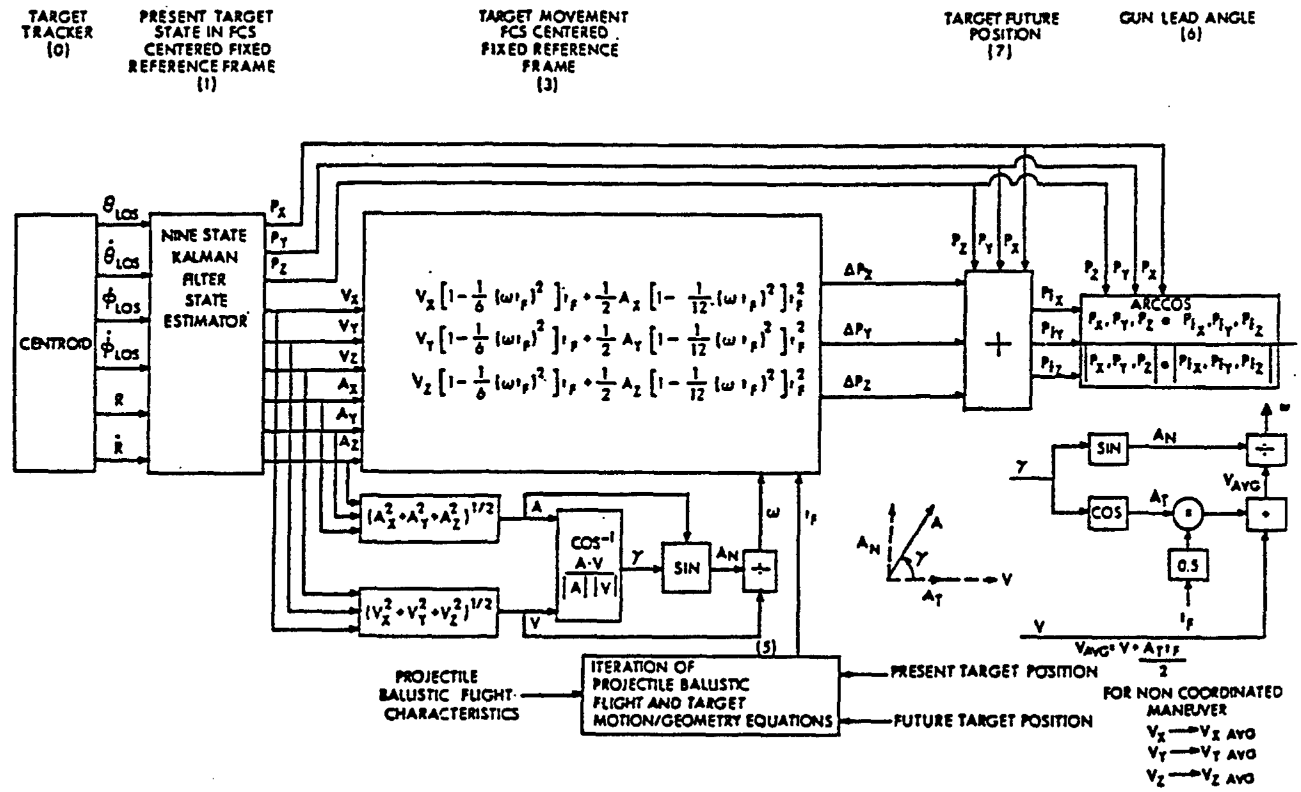


Figure 16.9 Generalized Description of 3D Non Linear CAAM Prediction FCS. (Alternate Method)

received the most attention by both analysts and system designers. For applications such as gun fire control systems, it is usually believed that the estimation problem should be closely tied to the prediction problem. The prediction process is open loop in nature, in that it generally has no feedback loops. Some systems, however, have closed projectile position feedback loops. One reason for more emphasis being placed on target present state estimation is that the present state estimates are usually employed in feedback paths making the estimator part of a closed loop system. Missile systems possess this characteristic but do not in general require the open loop prediction solution.

The techniques used to model the prediction process combine the projectile time of flight with the target present state estimates to provide either linear (Vt_{of} +) or second order ($Vt_{of} + 1/2\ At_{of}^2$) prediction in each coordinate of the s-l-m reference frame. Variations of the second order predictor are sometimes employed to adjust the decay of A during the prediction interval, and are determined on an empirical basis. Another variation of the prediction process used is to model the acceleration as a first order Gauss-Markov process. Note that the t_{of} interval is much greater than the update intervals usually employed in the present state estimation process. The goal to integrate closely the gun fire control estimation-prediction processes is certainly a worthwhile objective, but the majority of efforts to date have not moved far enough away from the application of Kalman-filtering and

Gauss-Markovian processes for both estimation and prediction to adequately cope with realistic maneuvering targets. Once the nature of what the prediction process must accomplish is understood, it will become clear as to how to build a high fidelity predictor model and tell only a little "lie" to the mathematical model rather than a big one. There is clearly recognizable and acceptable rationale that supports selection of the s-l-m reference frame for the target present state estimator, because the Kalman filter's observation inputs are then straightforward measurements and the update interval for the filter is much greater than the target maneuver time constant. For the prediction process, it is logical to require that it use the target present state estimates in some manner to determine target future position a time of flight later.

The locus of points in three dimensional space that describe the path of a maneuvering target generates a complex curve. It is the characteristics of this curve that will fully describe the evolution of the trajectory of a maneuvering target during the time of flight of a projectile. Only two things have to be known about this complex curve to fully describe its trajectory; its curvature and torsion. These two characteristics, to be a meaningful description of the target's path, must be referenced to a target oriented reference frame, with its principal axis aligned with the target's velocity. Another axis, orthogonal to the velocity oriented axis, along with the velocity oriented axis, forms a plane, referred to as the maneuver plane. The orthogonal axis to the velocity forms the principal normal to the space curve and contains both the three dimensional curve and the arc of a circle at the tangency point where the maneuver plane touches the space curve. The third axis is referred to as the bi-normal and is perpendicular to the maneuver plane. Its turn rate about the principal axis produces what is referred to as torsion of the target aligned reference frame, giving a complex three dimensional curving nature to the maneuvering target's trajectory.

This model is a robust representation of the most generalized maneuvering in three dimensions that a target can execute. It therefore forms the basis for the design of a high fidelity gun fire control predictor. At the outset of the effort to develop such a prediction model, one observation is in order; the realistic prediction process being strived for will probably initially be constrained by the fact that the potential three dimensional curving nature of the path during the prediction interval will restrict acceptable prediction to regions where there is little or no torsion of the maneuvering plane.

REFERENCE

16.1 Burke, H.H., ed., Maneuvering Targets and Gun Fire Control Technology, GACIAC SOAR-92-01, IIT Research Institute, Chicago, IL, February 1993.

This page is intentionally blank.

CHAPTER 17

AN ASSESSMENT OF MODERN CONTROL THEORY

17.1 An Assessment of Air-to-Air Missile Guidance and Control Technology.

Applications of modern control theory to air-to-air operations push the state-of-the-art more than any other application. The air-to-air missile problem[17.1] involves the pursuit of a highly maneuverable aircraft by a tactical guided missile. This application of modern control technology requires the estimation of the target's motion in 3-D space, the implementation of a guidance law which generates optimal intercept steering commands, and the control of a dynamic system, the missile, modeled by a set of highly-nonlinear, time-varying, multivariable, coupled, uncertain differential equations. The overall problem can be subdivided into three general subproblems involving estimation, guidance, and control. These three subproblems remain nonlinear and time-varying and when combined to construct the overall mathematical model of the problem, result in a complex integrated system of equations.

Figure 17.1 is a generic block diagram of an advanced air-to-air missile guidance and control system based on modern control technology. The seeker section contains a sensing mechanism which generates a stream of information about the target aircraft. This information stream is processed in numerical form by a target state estimator such as an extended Kalman filter. This filter, or estimator, provides an indication of the relative target-to-missile position, velocity, and acceleration. The nature of these estimates depends on the model assumed for the target's acceleration. A guidance law based on optimal control theory operates on the target state estimates and an auxiliary estimate of the time-to-go until intercept and produces acceleration commands indicating the required motion of the missile. The autopilot converts these acceleration commands into actuator commands which in turn repositions the missile's control surfaces. The actuator commands are based on the missile airframe's aerodynamic characteristics, the sensed missile body angular velocities, and the sensed missile linear accelerations. The deflections of the control surfaces cause changes in the missile's dynamic state, and these changes close the three feedback loops indicated in the figure.

Much basic research intended to improve air-to-air missile guidance and control has been conducted over the past 15 years. This research has included work on target state estimation, target acceleration modeling, target tracking, target maneuver detection, guidance law development, and bank-to-turn autopilot design. Techniques of modern control theory investigated in these efforts have included adaptive filtering, nonlinear filtering, parameter identification, optimal control, state-variable methods, adaptive control, dual control, and differential game theory.

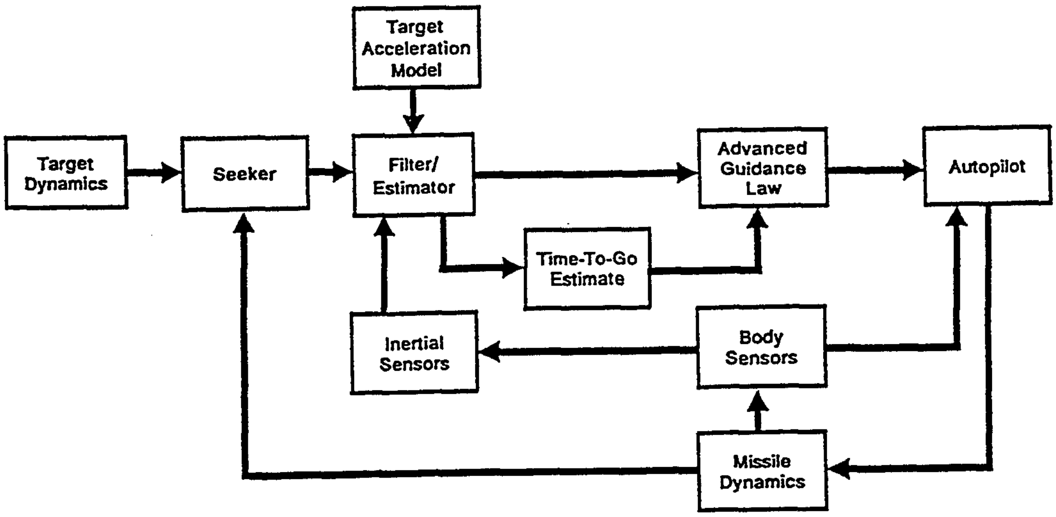


Figure 17-1. Block diagram components of a guided missile

17.2 Target State Estimation

Target state estimation involves the use of a state variable filter, or state variable estimator, to estimate the present state of the target. The design of the estimation algorithm depends on the mathematical model selected for the motion of the target. This model can vary in complexity from a simple point-mass object moving in a constant direction at a constant velocity to a complicated dynamic model which includes the target orientation and aerodynamics. The purpose of the target state estimator is to generate a stream of reliable data useful for target tracking and target maneuver detection.

Various filtering techniques have been used in the historical development of target acceleration models to form effective target trackers. Merging a point or jump process with a continuous-time Gaussian process, either additively or by parametric imbedding, was an excellent way to model a highly-maneuverable target's acceleration. Of the several models investigated, only one gave any consideration to the target's aerodynamics. It was suggested that a natural next step in this area is the development of a target acceleration model which merges the aerodynamic characteristics of the target with a correlated acceleration process implemented by a Gauss-Markov model. In such a model a set of aerodynamic parameters would correspond to a specific target maneuver and an abrupt change in the parameter set would correspond to the initiation of a new maneuver.

17.3 State Variable Filtering and Estimation Techniques

The Kalman filter, the extended Kalman filter, and a variety of other stochastic filters have been used to estimate state variables of dynamic systems corresponding to target motion models. Various filtering techniques have been used in conjunction with one or more target motion models. The objective is to identify the most effective filters. The design of any state variable estimator depends on the mathematical model used to define the underlying dynamic system.

Only two filters which have been observed employed a spherical coordinate system, and these two filters produced simpler and more accurate trackers than the corresponding filters based on a Cartesian coordinate system. Since the target dynamics are nonlinear in either a spherical or a Cartesian coordinate system, linearizing nonlinear spherical target dynamics should be no more detrimental than linearizing Cartesian target dynamics. There is no compelling reason to model the target's acceleration in the Cartesian coordinate system. By developing linear models of inertial radial and angular acceleration directly in spherical coordinates, the entire target state variable estimation process could be performed directly in the spherical coordinate system.

A filter implementation of this type would require a nonlinear transformation of inertial strapdown outputs to spherical coordinates and an inverse nonlinear transformation of the filter's output to Cartesian coordinates. In a dual control application this nonlinear transformation of state variables could be avoided if the guidance law were also formulated in spherical coordinates. A research effort is required to investigate the best way to model radial and angular target acceleration and to determine a simple way to incorporate target aerodynamic characteristics into that model. One particular application where spherical coordinates work well is for applications of modern control theory to strategic missiles. Tactical missiles rely on a flat-Earth, constant gravity model. When missile ranges exceed 100 miles, this model is inaccurate. However, all of the tools of modern control theory can be adapted to the strategic mission. Zarchan [17.2] provides a discussion of the differences between tactical and strategic missile control and includes some numerical solutions.

Extended Kalman filters and other nonlinear filters have shown marked improvement over the standard Kalman filter in many applications, but have shown only limited effectiveness as air-to-air target trackers. This lack of effectiveness is probably due to the inadequacy of the underlying target acceleration models. The use of an adaptive filter capable of responding to rapidly changing target motions is suggested. The use of a multi-model Kalman filter is also suggested as being able to outperform its single-model counterpart.

17.4 Advanced Guidance Laws

Three guidance phases—midcourse, terminal, and endgame—are involved in the air-to-air engagement. Midcourse guidance begins at the time of launch and lasts until the time of seeker acquisition of the target. During midcourse guidance an on-board inertial navigation system provides estimates of missile position, velocity, and acceleration. The launch aircraft may, in a nonautonomous system, provide a periodic estimate of the target's position and velocity. Once the missile seeker acquires the target, the midcourse guidance phase ends and the terminal guidance phase is initiated. An active seeker (e.g., radar) can provide noisy measurements of the line-of-sight angle to the target, the target range, and the range rate. The final seconds of terminal guidance comprise the endgame, and this time interval is often treated as a separate guidance phase. The reason for this special treatment is that target maneuvers can be most effective at that time. A well-timed target maneuver during the endgame increases the probability that the target can defeat the guidance law. This result comes about due to the finite response time of the missile airframe response (typically 0.25 to 0.50 seconds) and the finite response time of the target state estimator (typically 0.50 seconds for a typical extended Kalman filter).

Linear-quadratic and nonlinear guidance laws based on modern control theory have been proposed for use during midcourse guidance. Performance measures considered by various researchers have included minimum kinetic energy loss, minimum time, and minimum heading error. Most research to date has focused on deterministic optimal control formulations of the midcourse guidance problem.

The optimal guidance of pulse motor missiles has been suggested as a worthy area of research. The fundamental problems associated with boost-sustain midcourse guidance appears to have been solved. The remaining issues involve algorithm implementation and problem formulation and solution. Algorithm implementation is concerned with software design methodology selection, high-order language run time characteristics, cross compiler efficiencies and hardware throughput, and memory limitations. Problem formulation and solution involve a choice of either a closed-form solution to an approximate optimal control formulation described by either a linear-quadratic-Gaussian or linear-quadratic-regulator problem, or an approximate numerical solution to an exact nonlinear optimal control problem. Analytic solutions to most nonlinear optimal control problems cannot be achieved due to the complexity of the nonlinear two-point boundary value problem, and numerical solution of the exact nonlinear optimal control problem is usually impractical in terms of computational burden.

Since seeker measurements are known to degrade considerably in the endgame, deficiencies in missile kill effectiveness are primarily associated with that phase, and to a certain extent with the terminal guidance phase as well. As seeker measurements degrade, a lack of target information results. A stochastic game approach may be a possible remedy for the lack of target information which occurs as seeker measurements degrade. A dual control approach is recommended as a possible solution to the reduction in target information available from endgame seeker measurements when using homing guidance.

Further research into guidance law and autopilot interactions should be pursued. The acceleration limits of the missile airframe and the autopilot's finite response time should be considered in these efforts. Imbedding of the autopilot dynamic model in the derivation of the guidance model is suggested as a possible approach.

17.5 Bank-to-Turn Autopilot Design

The design of autopilots for missiles having a symmetrical airframes and skid-to-turn control schemes is now a relatively mature application of classical control theory. Recent interest in the development of missiles having non-symmetrical airframes and bank-to-turn control schemes has stimulated interest in the design of autopilots for these missiles. The most prominent missile to use bank-to-turn is the Tomahawk cruise missile. The use of a non-symmetrical airframe cross-section is known to improve the aerodynamic efficiency of both the missile and the launch aircraft. The non-symmetrical cross-section gives the missile a large pitch-plane acceleration capability and a corresponding small yaw-plane acceleration capability. Rolling or banking is thus required to maximize the missile's maneuverability. Engine performance constraints and other factors require the missile angle of attack to remain positive and the missile side-slip angle to remain small. The non-symmetrical missile cross-section results in a highly-nonlinear system of dynamic equations which determine the angular motions about the roll, pitch, and yaw axes.

Classical control theory, linear quadratic Gaussian regulator theory, and eigenvalue assignment methods have all been used to design bank-to-turn autopilots. In a classical approach, the design of an autopilot is based on a linear missile dynamic model assumed to be valid in the neighborhood of a designer-selected operating point. The initial design ignores the dynamic interaction between the three axes, and allows a set of three single-axis controllers to be developed. For a bank-to-turn missile, the initial control system parameters are selected so that the missile response time requirements are satisfied by the pitch and roll axes controllers, and the controlled yaw axis response is required to be at least as fast as the response of the roll axis.

Airframe control using a classical bank-to-turn autopilot is obtained by operating the yaw axis controller as a regulator which minimizes the side-slip angle. The pitch and roll axis acceleration commands generated by the guidance system are passed directly to the pitch and roll axis autopilots. If necessary, overall performance can be improved by the feedback of pitch and yaw axis accelerations to the pitch and yaw autopilots and simultaneously providing a feedback of roll rate to the pitch and yaw channels. The control system gains generally vary as a function of dynamic pressure and this partially accounts for aerodynamic parameter variations over the desired missile operating range. Full six-degree-of-freedom simulations followed by extensive flight testing are used to evaluate and finalize the autopilot design.

The design of an autopilot design based on the application of linear-quadratic-Gaussian regulator theory begins by constructing the nonlinear mathematical model of the airframe's dynamics. In one approach[17.3] the model was decoupled into two submodels—a roll axis model and a composite pitch and yaw axis model. Roll rate was treated as an external input to the composite pitch and yaw axis model. These nonlinear models were linearized over the specified aerodynamic operating region. A constant-gain, reduced-order Kalman filter was employed to estimate the unmeasured state variables and permitted implementation of a closed-loop feedback controller. The separation principle was applied, and the required controller gains were determined by using a combination of several design methods. A pole-placement procedure was used to obtain the necessary roll axis time response. Pitch and yaw axis controller gains were determined by applying linear-quadratic-Gaussian theory with loop-transfer-recovery. The side-slip angle was minimized by including it in the performance measure and applying a large weighting value. By treating the guidance system command accelerations as state variables of the autopilot, integral control action was obtained. In the final design the controller gains were scheduled as a function of dynamic pressure and roll rate.

Pole-placement methods which permit the simultaneous placement of system eigenvectors and eigenvalues have been applied to the design of autopilots for aircraft applications. This method relies on a multi-input, multi-output state variable model for the autopilot and the controlled dynamic system, and yields the ability to decouple the interacting control channels. This approach may also be useful for the design of autopilots for a highly-coupled bank-to-turn system.

In general, a properly designed autopilot developed by either classical or modern control theory means can be expected to perform reasonably well. However, when the design of an autopilot is based on a multi-input, multi-output mathematical model, the classical design approaches are cumbersome and require significant approximations to be made in order to yield a solution to the control problem. Design approaches based on modern control theory are more likely to be limited by the availability of suitable hardware. At present, a number of simplifying assumptions including the use of a low-order model, a constant roll rate, a reduced order and constant-gain state variable estimators and simplified controller gain scheduling techniques, were required to design and implement an autopilot based on modern control theory that would fit into a current generation, tactical missile microcomputer.

The linear-quadratic-Gaussian design approach provides the control system designer with a means for including the effects of the full set of state and control variables in the performance measure. The relative effect of each variable can be adjusted by means of its weighting factor. The pole-placement eigen-structure approach provides the designer with a means to select and assign values to the frequency domain parameters of damping factor and natural frequency. Robustness is achieved in each of these approaches by a different effect. In the linear-quadratic-Gaussian approach, design robustness is achieved via loop transfer recovery. In the eigen-structure approach, robustness is achieved by decoupling of the control loops. A unified design procedure is required to merge these time and frequency approaches.

In the design of bank-to-turn (and other) autopilots, a number of design issues remain unresolved. These include the form of the nonlinear dynamic system model, methods for linearization and model order reduction, the assessment of sensitivity and robustness, the role of adaptive control, the effect of digital controller implementations, the use of special-purpose computer architectures for high-speed fault-tolerant computation and control, and the integration of the guidance and control problems into a unified control strategy.

17.6 Summary

The current goals for modern control theory sound like more of the same: develop new non-linear optimum guidance laws; improve target estimation filter structures; integrate state estimators, guidance laws, and autopilot design; operate at higher angles of attack, optimize algorithms, and, provide prompt reprogrammability. Nevertheless, these are the objectives demanded for highly maneuverable precision guided missiles to intercept cruise missiles, ballistic missiles, and highly maneuverable airborne platforms. The conventional design approach is indicated in Figure 17.2. In this approach, it is necessary to merge multiple independent designs. Performance is constrained by disturbance effects and model uncertainties. The trend in current control system designs is to integrate as many functions as possible as symbolized in Figure 17.3. The objective is the application of H_∞ optimal control and estimation theory in a single closed-loop system. The interactions of various control functions are coupled to minimize the effects of external disturbances and model uncertainties.

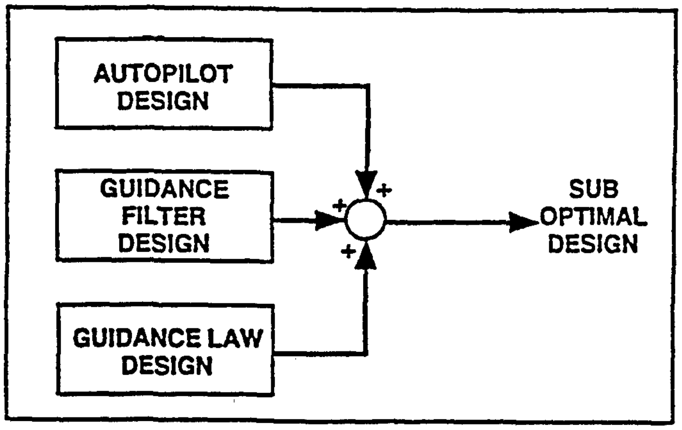
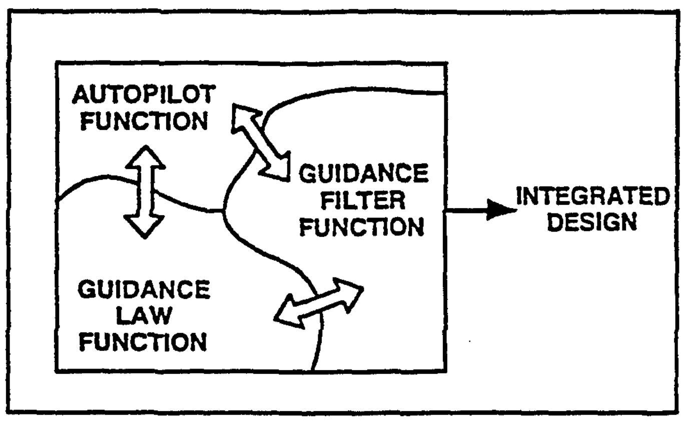


Figure 17.2. Conventional control system design



17.3 Use of an integrated design in advanced control systems

Several new technologies are driving the advancement of control technology. Guidance, navigation and control components have greater capabilities, are more compact, weigh less and may even eventually cost less. Inertial measurement units are becoming more precise in small packages. GPS applications are proliferating. Anti-jamming capabilities are being included with GPS. Digital processors, neural net algorithms, fuzzy logic, and wavelet tools stretch the options for innovative designs for missile and submunition components. Surface and undersurface guidance and control is also advancing, as well as gun fire control.

REFERENCES

17.1 Cloutier, J.R. and Evers, J.H. Air-to-Air Missile Guidance and Control Technology, Tutorial Workshop on Advanced Navigation, Guidance and Control, and Their Applications; 29th IEEE Conferences on Decision and Control, Honolulu, Hawaii, December 3-4, 1990.

17.2 Zarchan, P., Tactical and Strategic Missile Guidance, Vol. 124, Progress in Astronautics and Aeronautics, American Institute of Aeronautics and Astronautics, Inc., Washington, DC, 1990.

17.3 Williams, D.E., Friedland, B., and Madiwale, A.N., Bank-to-Turn Autopilot Design Using Modern Control Theory, Proceedings of the Workshop on Bank-to-Turn Controlled Terminal Homing Missiles, GACIAC PR-85-01, pp.157-180, January 1985.

This page is intentionally blank.

GUIDANCE AND CONTROL
INFORMATION ANALYSIS CENTER (GACIAC)

GACIAC is a DoD Information Analysis Center operated by IIT Research Institute under the technical sponsorship of the Joint Service Guidance and Control Committee with members from OSD, Army, Navy, Air Force, and ARPA. The AMC Smart Weapons Management Office of the U.S. Army Missile Command provides the Contracting Officer's Technical Representative. GACIAC's mission is to assist the guidance and control community by encouraging and facilitating the exchange and dissemination of technical data and information for the purpose of effecting coordination of research, exploratory development, and advanced technology demonstrations. To accomplish this, GACIAC's functions are to:

1. Develop a machine-readable bibliographic data base -- currently containing over 45,000 entries;

2. Collect, review, and store pertinent documents in its field of interest -- the library contains over 20,000 reports;

3. Analyze, appraise, and summarize information and data on selected subjects;

4. Disseminate information through the GACIAC Bulletin, bibliographies, state-of-art summaries, technology assessments, handbooks, special reports, and conferences;

5. Respond to technical inquiries related to weapon guidance and control; and

6. Provide technical and administrative support to the Joint Service Guidance and Control Committee (JSGCC).

The products and services of GACIAC are available to qualified industrial users through a subscription plan or individual sales. Government personnel are eligible for products and services under block funding provided by the Army, Navy, Air Force and ARPA. A written request on government stationery is required to receive all the products as a government subscriber.

Further information regarding GACIAC services, products, participation plan, or additional copies of this state-of-the-art review may be obtained by writing or calling: GACIAC, IIT Research Institute, 10 West 35th Street, Chicago, Illinois 60616-3799; Area code 312, 567-4587; Fax 312, 567-4889.